Scalarizing Functions in Decomposition-based Multiobjective Evolutionary Algorithms

Shouyong Jiang, Shengxiang Yang, *Senior Member, IEEE*, Yong Wang, *Member, IEEE*, and Xiaobin Liu

***Abstract*—Decomposition-based multiobjective evolutionary algorithms have received increasing research interests due to their high performance for solving multiobjective optimization problems. However, scalarizing functions, which play a crucial role in balancing diversity and convergence in these kinds of algorithms, have not been fully investigated. This paper is mainly devoted to presenting two new scalarizing functions and analyzing their effect in decomposition-based multiobjective evolutionary algorithms. Additionally, we come up with an efficient framework for decomposition-based multiobjective evolutionary algorithms based on the proposed scalarizing functions and some new strategies. Extensive experimental studies have demonstrated the effectiveness of the proposed scalarizing functions and algorithm.**

***Index Terms*—Multiobjective optimization, scalarizing function, decomposition, evolutionary algorithm.**

I. Introduction

MULTIOBJECTIVE evolutionary algorithms (MOEAs) [6] have been shown to be well-suited for multiobjective optimization problems (MOPs) as they can approximate the Pareto-optimal front (PF) with a population in a single run. After decades of development, a number of MOEAs have emerged in the field of evolutionary multiobjective optimization (EMO). According to their selection techniques, these MOEAs can be generally grouped into three different classes: Pareto dominance based methods, such as the nondominated sorting genetic algorithm II (NSGA-II) [5] and strength Pareto evolutionary algorithm 2 (SPEA2) [51]; indicator-based methods, such as the indicator-based evolutionary algorithm (IBEA) [50]; and decompostion-based methods, such as the multiple single objective Pareto sampling (MSOPS) [13], cellular multiobjective genetic algorithm (C-MOGA) [30], and multiobjective evolutionary algorithm based on decomposition (MOEA/D) [46].

Manuscript received September 22, 2016; revised February 2, 2017 and April 28, 2017; accepted May 5, 2017. This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) of U.K. under Grant EP/K001310/1, the National Natural Science Foundation of China (NSFC) under Grant 61673331 and 61673397, and the EU Horizon 2020 Marie Sklodowska-Curie Individual Fellowships under Project 661327 (*Corresponding author: Shengxiang Yang*).

Shouyong Jiang is with the School of Computing Science, Newcastle University, Newcastle upon Tyne, NE1 7RU, U.K. (email: math4neu@gmail.com).

Shengxiang Yang is with the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort Univesity, The Gateway, Leicester LE1 9BH, U.K. (email: syang@dmu.ac.uk).

Yong Wang is with School of Information Science and Engineering, Central South University, Changsha 410083, China (email: ywang@csu.edu.cn).

Xiaobin Liu is with School of Information Science and Engineering, Northeastern University, Shenyang 110819, China (email: xbl0918@gmail.com).

Decomposition-based MOEAs are a popular class of metaheuristics for EMO. They decompose an MOP into a number of subproblems[1] and simultaneously solve them in a collaborative manner. MOEA/D [46] is a representative of this class of metaheuristics. MOEA/D decomposes an MOP by scalarizing functions (or termed decomposition approaches in some studies [46]) into a set of subproblems, each of which is associated with a search direction (or weight vector) and assigned a candidate solution. In every generation, parents from a mating pool are selected to generate an offspring solution for each subproblem. Then, the offspring replaces certain existing solutions if it achieves better scalarizing values. So far, there have been a number of contributions to the improvement of MOEA/D, with regard to the following aspects:

1) Weight vectors: The quality (particularly uniformity and coverage) of approximations depends largely on the chosen weight vectors. In the original MOEA/D, the simplex lattice design [4] was used to construct weight vectors. However, it was later found that this method cannot ensure the uniformity of the obtained solutions on the PF [32], and other methods, such as uniform design [38], weight transformation [32], and two-layered design [8], [25], were therefore introduced to reduce this drawback. It is noteworthy that the specification of weight vectors depends largely on PF shapes, and inappropriate weight vectors can lead to poor performance of decomposition-based approaches [17], [19].
2) Scalarizing functions: Scalarizing functions play a fundamental role in MOEA/D and its variants. They can significantly affect the search ability of the evolving population and the quality of the resulting approximations. It has been suggested that adaptive scalarizing functions are beneficial to balance diversity and convergence at different search stages [16], [39]. Some existing scalarizing functions have been adapted to improve the quality of solutions [9], [20], [34], [35], [40].
3) Mating selection: In MOEA/D, each subproblem requires a mating range where parents are selected to mate. As a result, mating selection has an important impact on population diversity and convergence. Originally, MOEA/D used subproblems' neighborhood as the mating range [46]. Later, Li *et al.* [26] studied the influence of the mating range and showed the benefit of using the whole population with a low probability. In [20], niching techniques were employed to determine

[1]Note that, subproblems can be not only single-objective optimization problems [46] but also multiobjective ones [28].

the mating range. The size of the mating range or neighborhood has been investigated in a number of studies [14], [48].

4) Genetic operators: MOEA/D needs appropriate genetic operators to generate promising offspring, depending on the difficulty of the problem to be optimized. In [26], two different genetic operators were investigated on complicated problems. In [10], cross entropy was successfully integrated into MOEA/D for continuous optimization. An adaptive variation operator [28] is used to solve hard-to-converge problems. Recently, Li *et al.* [27] has combined covariance matrix adaptation evolution strategy (CMA-ES) [12] and differential evolution (DE) [7] for biased multiobjective optimization. Additionally, adaptive strategies [23] have been developed to automate the selection of proper operators for MOEA/D.
5) Replacement selection: When an offspring solution is generated, MOEA/D needs to decide what kinds of old solutions to be replaced by the new solution and how to do the replacement. The original MOEA/D [46] defines a replacement range/neighborhood for each subproblem, and old solutions in this range will be replaced if they are no longer promising. However, it has been found that it is more efficient to induce replacement within the neighborhood of the offspring's most suitable subproblem [41]. Other promising methods include the adoption of effective replacement range [45] and the use of constrained replacement range [40][2]. Instead of considering a proper replacement range, some studies focus on the match between subproblems and solutions, and thus stable matching [25] and inter-relationship [24] based selection strategies are proposed to facilitate replacement. In addition, chain-reaction replacement strategies [36] also help to enhance replacement selection in some sense.
6) Resource allocation: In MOEA/D, different subproblems may have different optimization difficulties. Thus, for efficiency, it is desirable to allocate computational resources to subproblems according to their difficulties. Much effort has been made along this direction, resulting in a number of effective resource allocation strategies [2], [47], [49].

Despite plenty of advances, MOEA/D still receives increasing research interests. A particular research direction is concerning scalarizing functions, which have not been fully explored yet. In MOEA/D, the weighted sum (WS), weighted Tchebycheff (TCH) and penalty-based boundary intersection (PBI) are the top three commonly-used scalarizing functions. These scalarizing functions respectively have their own strengths and drawbacks [39], [43], [45]. In view of the advantages and disadvantages of each scalarizing function, Ishibuchi *et al.* [16], [18] proposed to use different scalarizing functions adaptively or simultaneously during the search. In [10], a generalized form of scalarizing functions was developed to cope with various PF geometries. Sato [35] proposed an inverted PBI method to overcome the poor spread performance of existing scalarizing functions in some problems. Modified or advanced scalarizing functions have also been developed to facilitate the environmental selection in other algorithms [3], [22], [34], [44].

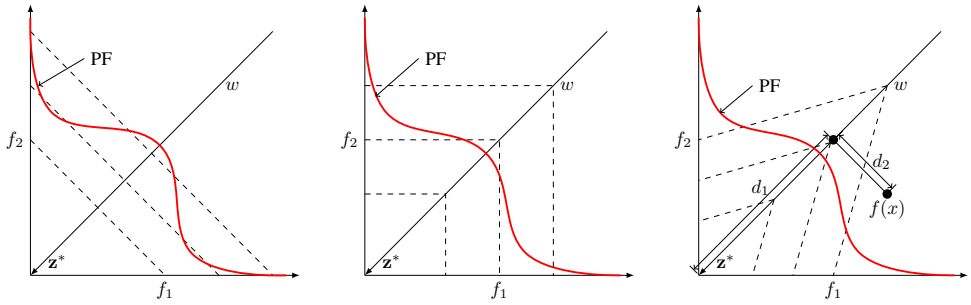
It is noteworthy that one important property of a scalarizing function is the shape or positioning of its contour lines [9], [43]. The contour lines are a set of equal scalarizing function values and play a crucial role in guiding the search in scalarizing search algorithms [9]. In [9], the authors argued that the dynamics of the search process is rather independent of the scalarizing function under consideration and instead mainly influenced by the induced contour lines. Another study [40] showed that imposing proper constraints to the contours of a scalarizing function can improve search efficiency. Recently, Wang *et al.* [39] have systematically studied the search ability of a family of widely-used scalarizing functions with different contours, called the L_p scalarizing functions, and have argued that different contours should be used at different search stages.

Generally, desired contours in scalarizing search algorithms can be obtained by 1) using traditional scalarizing functions and modifying their contours by adding constraints [40] or specifying different parameters [39], or 2) designing new and effective scalarizing functions. The former method looks intuitive and easy but may not always generate the exact contours one wants. For example, the modified scalarizing functions (excluding PBI) in both [40] and [39] cannot produce contour lines that have opening angles [9] smaller than $\pi/2$, which may not balance diversity and convergence well during the search. In this paper, the focus is on the latter, and new scalarizing functions that can induce adjustable contour lines are presented. The main contributions of this paper are summarized as follows.

- Two new scalarizing functions with adjustable contours are proposed and their properties are analyzed.
- On the basis of the new scalarizing functions and other new techniques, an efficient MOEA/D framework, called eMOEA/D, is introduced. eMOEA/D uses adaptive scalarizing functions to guide the search and a new replacement strategy to efficiently update the population.
- The effectiveness of the new scalarizing functions is verified. Extensive algorithm comparisons and discussions are conducted, and experimental studies show that eMOEA/D obtains better performance than the other compared algorithms.

The rest of this paper is organized as follows. Section II describes some background knowledge. Section III presents the proposed scalarizing functions and corresponding theoretical analysis, followed by preliminary experiments with regard to their influence in Section IV. Section V presents the efficient MOEA/D framework based on the scalarizing functions and some new techniques. Algorithm comparisons are provided in Section VI, followed by extensive discussions in Section VII. Section VIII concludes this paper.

[2]Note that, adding constraints to subproblems in [40] is actually equivalent to restricting replacement range.



(a) WS (b) TCH (c) PBI

Fig. 1: Illustration of three scalarizing functions on weight vector w, where dashed lines are contour lines.

II. Related Work

A. Basic Concepts

An MOP can be mathematically described as follows:

$$\begin{aligned} \min \quad & f(x) = \Big(f_1(x), f_2(x), \ldots, f_m(x)\Big)^T \\ s.t. \quad & x \in \Omega_x \end{aligned} \tag{1}$$

where $\Omega_x \subseteq \mathbb{R}^n$ is the decision space and $x = (x_1, \ldots, x_n)^T$ is a candidate solution. $f : \Omega_x \mapsto \Omega_f \subseteq \mathbb{R}^m$ contains m objective functions, and Ω_f is the attainable objective space.

Definition 1. *A solution x is said to dominate another solution y if x is not worse than y in all objectives and is better than y in at least one objective. This is denoted as $x \prec y$.*

Definition 2. *A solution x^* is said to be Pareto optimal if no another solution x in the decision space satisfies $x \prec x^*$.*

Definition 3. *The Pareto-optimal set (PS) is a set of Pareto-optimal solutions, i.e., $PS = \{x \in \Omega_x | x \text{ is Pareto optimal}\}$. Correspondingly, the image of PS in the objective space is called the Pareto-optimal front (PF), i.e., $PF = \{f(x) \in \Omega_f | x \in PS\}$.*

Definition 4. *For a given scalarizing function SF, a solution x is said to be better than another solution y if x obtains a better SF value than y. This is denoted as $x \prec_{SF} y$.*

Definition 5. *For a given SF, the improvement region (in the objective space) of a solution x is denoted as $\Phi(x) = \{f(\bar{x}) \in \Omega_f | \bar{x} \prec_{SF} x\}$.*

B. Scalarizing Methods in MOEA/D

1) WS Method: Assume that $w = (w_1, \ldots, w_m)^T$ is a weight vector where all components are non-negative and should satisfy $\sum_{i=1}^{m} w_i = 1$. The WS method defines the following single-objective problem:

$$\begin{aligned} \min \quad & g^{ws}(x|w, z^*) = \textstyle\sum_{i=1}^{m}(w_i|f_i(x) - z_i^*|) \\ s.t. \quad & x \in \Omega_x. \end{aligned} \tag{2}$$

If necessary, throughout the paper, $f_i(x) - z_i^*$ should be replaced by $(f_i(x) - z_i^*)/(z_i^{nadir} - z_i^*)$ where z_i^* and z_i^{nadir} are the i-th objective values of ideal point and nadir point found so far [46], respectively. The WS method is illustrated in Fig. 1(a), and it can obtain a set of PF points by different weight vectors. The method can approximate the PF if it is convex, but will miss some PF points if the PF is nonconvex [46].

2) TCH Method: The TCH method transforms an MOP into a scalar problem in the following form:

$$\begin{aligned} \min \quad & g^{te}(x|w, z^*) = \max_{1 \le i \le m} \Big(\tfrac{1}{w_i}|f_i(x) - z_i^*|\Big) \\ s.t. \quad & x \in \Omega_x, \end{aligned} \tag{3}$$

where $w_i = 10^{-4}$ is used in this method if $w_i = 0$. In Eq. (3), $1/w_i$ instead of w_i is adopted in order to obtain a set of uniformly-distributed solutions from a set of uniformly-distributed weight vectors [25]. The TCH method has the advantage in approximating nonconvex PFs compared with the WS method. It has been widely employed as a decomposition approach in MOEA/D variants [20], [25], [26], [40]. This method is shown in Fig. 1(b).

3) PBI Method: The PBI method transforms an MOP into a scalar problem as follows:

$$\begin{aligned} \min \quad & g^{pbi}(x|w, z^*) = d_1 + \theta d_2 \\ s.t. \quad & x \in \Omega_x, \end{aligned} \tag{4}$$

where

$$d_1 = \frac{\|(f(x) - z^*)^T w\|}{\|w\|}, \tag{5}$$

$$d_2 = \|f(x) - (z^* + d_1 \frac{w}{\|w\|})\|. \tag{6}$$

In PBI, θ is a user-defined penalty factor. d_1 and d_2 are the length of the projection of vector $(f(x) - z^*)$ on the weight vector w and the perpendicular distance from $f(x)$ to w, respectively. Fig. 1(c) provides a brief illustration of the PBI approach. It is clear that θ is a key parameter for balancing convergence (measured by d_1) and diversity (measured by d_2). Recent studies [15], [35] have shown that, when PBI approximates convex PFs, large diversity is likely to be obtained from minute and large θ values, and small θ values are beneficial to convergence.

C. Motivation

Scalarizing functions (or decomposition approaches) play a fundamental role in the performance of decomposition-based MOEAs. Scalarizing functions and decomposition-based MOEAs work closely as follows. First, scalarizing functions

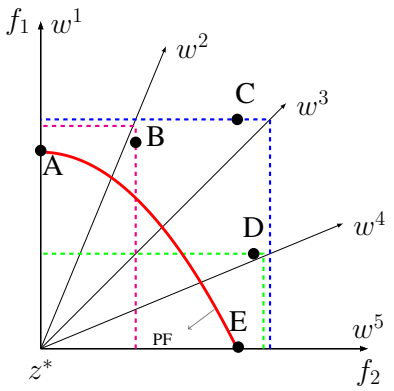


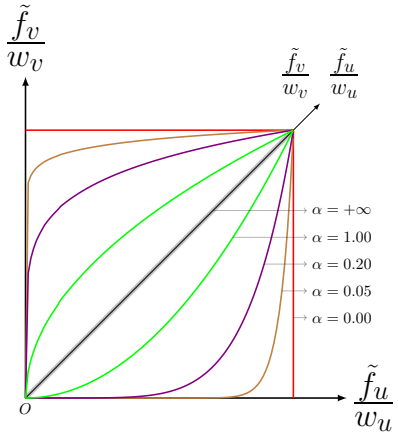
Fig. 2: Illustration of solution distribution in the bi-objective space. Dashed lines are contours of L_∞ scalarizing functions.

transform an MOP into a number of single-objective optimization problems. Then, decomposition-based MOEAs optimize each subproblem in a collaborative manner. If an inappropriate scalarizing function is chosen or the used scalarizing function cannot transform MOPs well to subproblems, then decomposition-based MOEAs may fail to approximate the PF.

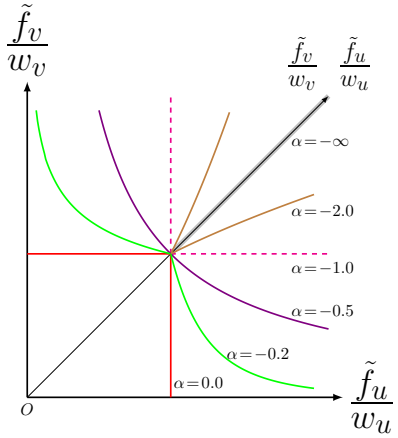
The three scalarizing functions mentioned previously, i.e., WS, TCH, and PBI, have been widely used in decomposition-based MOEAs. Despite their great success in solving a variety of MOPs, these scalarizing functions have their own limitations. For example, PBI is very sensitive to the search landscape of the objective space and may miss some PF points if the underlying penalty factor is not well-tuned [35], [43].

On the other hand, WS and TCH belong to the family of the L_p ($p \geq 1$) scalarizing functions [39], and they are two extreme cases of this family (WS and TCH correspond to L_1 and L_∞, respectively). As pointed out by Wang *et al.* [39], TCH is the best in the L_p family in terms of diversity maintenance because its contour lines have the smallest opening angle, i.e., $\pi/2$, as shown in Fig. 1(b). However, we argue here that a contour with $\pi/2$ opening angle is still insufficient to maintain diversity in some cases. Fig. 2 illustrates a situation where the L_∞ scalarizing function fails to maintain diversity. In the figure, three individuals (B, C and D) are of great importance to diversity, but they will be replaced by two boundary individuals (A and E) because their improvement regions contain A and/or E, i.e., $\{A\} \subset \Phi(B)$, $\{A, E\} \subset \Phi(C)$, and $\{E\} \subset \Phi(D)$. This implies that L_∞ along with its family members lacks the property of maintaining/promoting diversity. It may be of little use and even fails if search environments are very complex and little information is available in advance.

Generally, the above-mentioned drawback with regard to L_p scalarizing functions can be alleviated by the following possible ways. First, constraints can be added to the contour lines of L_p scalarizing functions to reduce opening angles or L_p scalarizing functions work with appropriate replacement strategies to stop abandoning diverse individuals. Second, a new and effective scalarizing function with adjustable contours or improvement regions that well control diversity would help MOEA/D yield good performance. Obviously, the second approach is more straightforward and easier to implement because it does not require any modification of the basic framework of MOEA/D. For this reason, it is desirable to devise new scalarizing functions for MOEA/D.



(a) $\alpha \geq 0$



(b) $\alpha \leq 0$

Fig. 3: Contour lines of MSF with different α values.

III. Proposed Scalarizing Functions

In the following, we propose two scalarizing functions for MOEA/D.

A. Multiplicative Scalarizing Function (MSF)

The MSF method is defined as follows:

$$g^{msf}(x|w, z^*) = \frac{\left[\max\limits_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x) - z_i^*|\right)\right]^{1+\alpha}}{\left[\min\limits_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x) - z_i^*|\right)\right]^{\alpha}} \quad (7)$$

where w_i is set to 10^{-4} if w_i equals zero, and α is a control parameter. If the denominator is zero, it is replaced with a minute positive value (e.g., 10^{-5}) to keep the division legal. When $\alpha = 0$, $g^{msf}(x|w, z^*)$ degenerates to Eq. (3). That is, the TCH method is a special case of MSF. MSF is technically able to find all Pareto optimal solutions and this is presented in the supplementary material.

For the convenience of denotion, let $\tilde{f}_i = f_i - z_i^*$. We also assume u and v ($1 \leq u, v \leq m$) are the indices that

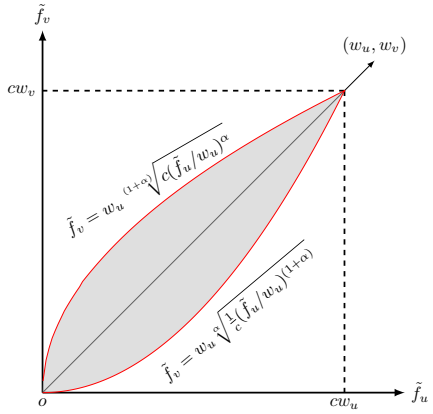


Fig. 4: The improvement region (shaded area) of MSF.

maximize and minimize $\tilde{f}_i/w_i$ $(1 \leq i \leq m)$, respectively. Fig. 3 presents contour lines of MSF with different α values on the $(\tilde{f}_u/w_u)(\tilde{f}_v/w_v)$-plane. As can be seen from the figure, the size of the improvement region decreases with the increase of α. Obviously, $\alpha \geq 0$ is more desired compared with $\alpha < 0$ because just like the WS method, MSF with $\alpha < 0$ is unable to approximate non-convex PFs and is very likely to lose diversity as mentioned earlier[3]. For this reason, we only consider $\alpha \geq 0$ in this paper.

Theorem 1. *On the $\tilde{f}_u\tilde{f}_v$-plane, the maximum size of the improvement region enclosed by $g^{msf}(x|w,z^*) = c$ $(c \geq 0)$, denoted $\Delta(c)$, is equal to $w_u w_v c^2/(2\alpha+1)$.*

Proof: The improvement region is enclosed by the curve $g^{msf}(x|w,z^*) = c$, as shown in Fig. 4. It is easy to see that $g^{msf}(x|w,z^*) = c$ and $\tilde{f}_u/w_u = \tilde{f}_v/w_v$ have two common points of intersection, i.e., (0,0) and (cw_u, cw_v). Next, we calculate the two parts of $g^{msf}(x|w,z^*) = c$ below and above $\tilde{f}_u/w_u = \tilde{f}_v/w_v$, respectively.

If $\tilde{f}_u/w_u > \tilde{f}_v/w_v$, then $g^{msf}(x|w,z^*) = (\tilde{f}_u/w_u)^{1+\alpha}/(\tilde{f}_v/w_v)^\alpha = c$. Thus, $f_v = w_v \sqrt[(1+\alpha)]{c(\tilde{f}_u/w_u)^\alpha}$.

If $\tilde{f}_u/w_u < \tilde{f}_v/w_v$, then $g^{msf}(x|w,z^*) = (\tilde{f}_v/w_v)^{1+\alpha}/(\tilde{f}_u/w_u)^\alpha = c$. Thus, $f_v = w_v \sqrt[\alpha]{\frac{1}{c}(\tilde{f}_u/w_u)^{(1+\alpha)}}$.

Therefore, $\Delta(c)$ is equal to the area bounded by the above two function curves. That is,

$$\begin{aligned}\Delta(c) &= \int_0^{cw_u} \left(w_v \sqrt[\alpha]{\frac{1}{c}(\frac{\tilde{f}_u}{w_u})^{(1+\alpha)}} - w_v \sqrt[(1+\alpha)]{c(\frac{\tilde{f}_u}{w_u})^\alpha} \right) d\tilde{f}_u \\ &= \frac{w_u w_v c^2}{2\alpha+1}.\end{aligned} \tag{8}$$

■

Theorem 2. *For m $(m \geq 2)$ objectives, the maximum size of the improvement region enclosed by $g^{msf}(x|w,z^*) = c$ $(c \geq 0)$, denoted $\Delta_m(c)$, is equal to*

$$\Delta_m(c) = \begin{cases} \frac{w_1 w_2 c^2}{2\alpha+1} & if\ m=2, \\ \frac{\left(\frac{c^2\pi(m-1)}{m}\right)^{\frac{m-1}{2}} \left(\prod_{i=1}^{m} w_i\right)(m-1)!}{\Gamma(\frac{m-1}{2}+1)(m(m\alpha+1)\cdots(m\alpha+(m-1)))} & otherwise, \end{cases} \tag{9}$$

where $\Gamma(\cdot)$ is the gamma function [31].

The proof of Theorem 2 is provided in our supplementary material. It is clear from the theorem that, the size of the improvement region for a predefined weight vector is controlled by α. When α is fixed, the improvement region is affected by the weight vector values, as shown in Fig. 5. Generally, intermediate weight vectors tend to have larger $\Delta_m(c)$ values than boundary ones. This is because the product of the elements of an intermediate weight vector is larger than that of a boundary weight vector. This means that subproblems associated with intermediate weight vectors have more opportunities to be updated than those with boundary weight vectors. Thus, to be fair for all the subproblems, it is plausible to penalize intermediate subproblems or compensate boundary subproblems so that all subproblems can have improvement regions with relatively equal sizes.

According to Theorem 2, a straightforward way of balancing subproblems is to vary α for different subproblems. In this paper, we adjust α as follows:

$$\alpha = \beta\{m \min_{1\leq i\leq m}(w_i)\} \tag{10}$$

where w_i is the ith element of weight vector w, and m is the number of objectives. $0 \leq m \min_{1\leq i\leq m}(w_i) \leq 1$ always holds under the assumption $\sum_{i=1}^{m} w_i = 1$. β is the underlying control parameter. It is worth mentioning that the improvement region in TCH is non-adjustable as it is constant (i.e., $\Delta_2(c) = w_1 w_2 c^2$) if both the contour line value c and the weight vector w are predetermined. However, MSF can control each subproblem's improvement region through the adjustment of α (or β in Eq. (10)).

B. Penalty-based Scalarizing Function (PSF)

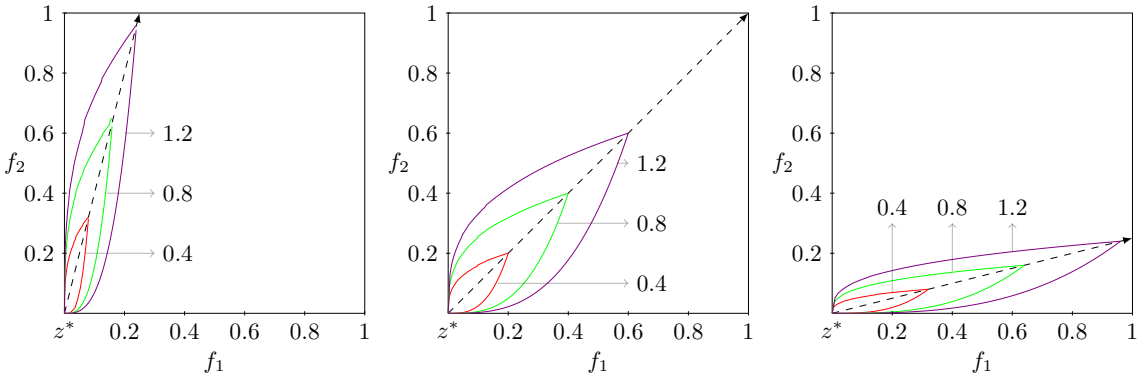
Inspired by the idea of PBI that controls diversity by penalizing solutions far from a weight vector, we modify the weighted Tchebycheff function in the following way:

$$g^{psf}(x|w,z^*) = \max_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x) - z_i^*|\right) + \alpha d, \tag{11}$$

$$d = \frac{\sqrt{\|f(x)-z^*\|^2\|w\|^2 - \|(f(x)-z^*)^T w\|^2}}{\|w\|}, \tag{12}$$

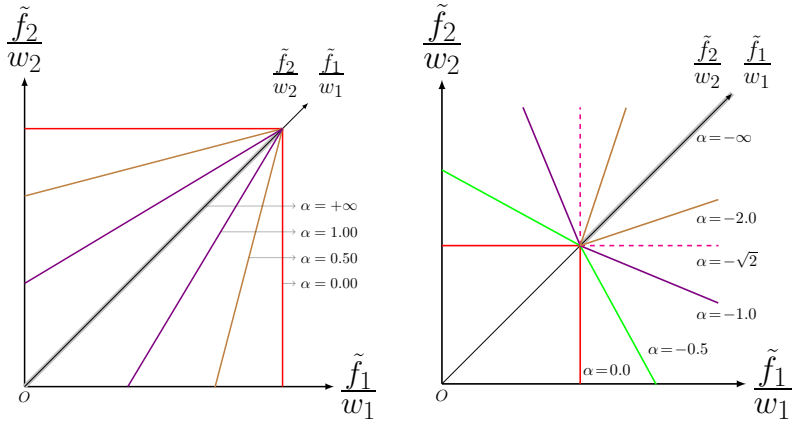
where d is the perpendicular distance of a solution to the weight vector w, i.e., d is the same as d_2 defined in Eq. (6). α is a penalty value used to control diversity. Fig. 6 illustrates the contour lines of PSF with different α values. Similar to MSF, the improvement region of PSF varies dramatically with α, and $\alpha \geq 0$ is preferable in this paper as diversity is the focus of this work. Fig. 7 presents the contour lines of PSF for different weight vectors. PSF is technically able to

[3]However, $\alpha \leq 0$ may provide fast convergence in the early stage of the search and solve concave MOPs efficiently.



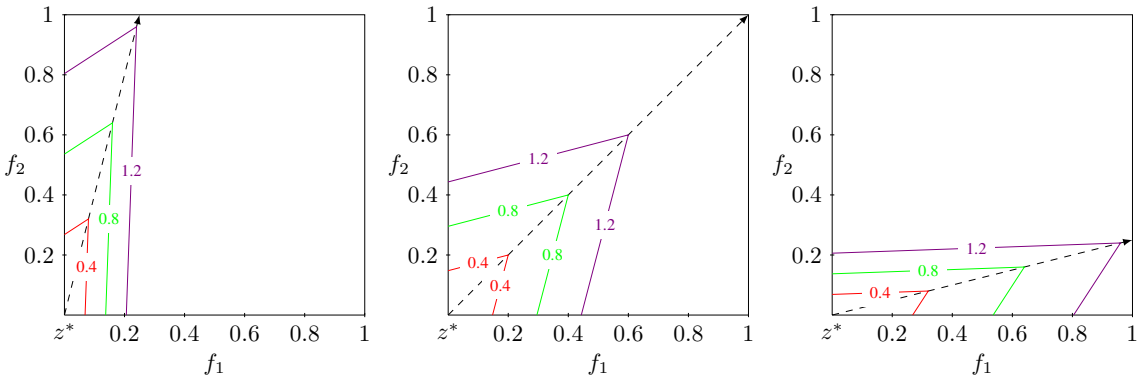
(a) $w = (0.2, 0.8)^T$ (b) $w = (0.5, 0.5)^T$ (c) $w = (0.8, 0.2)^T$

Fig. 5: Contour lines of MSF with $\alpha = 0.5$ for the contour values 0.4, 0.8, and 1.2.



(a) $\alpha \geq 0$ (b) $\alpha \leq 0$

Fig. 6: Contour lines of PSF with different α values.



(a) $w = (0.2, 0.8)^T$ (b) $w = (0.5, 0.5)^T$ (c) $w = (0.8, 0.2)^T$

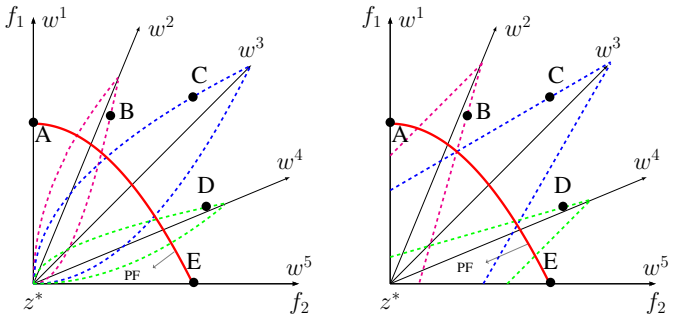
Fig. 7: Contour lines of PSF with $\alpha = 1.0$ for the contour values 0.4, 0.8, and 1.2.

find all Pareto optimal solutions and this is presented in the supplementary material.

Like MSF, the size of the improvement region of PSF also depends largely on α and the considered weight vector w. Thus, PSF use the same adjustment strategy (see Eq. (10)) to balance different subproblems.

C. Remarks

1) MSF and PSF vs L_p: Let us revisit the problem previously illustrated in Fig. 2, where L_∞ and other L_p scalarizing functions cannot induce proper contours or improvement regions to avoid diversity loss. We wonder whether the proposed MSF and PSF can overcome this drawback. Fig. 8 presents the contours of MSF and PSF (both are with $\alpha = 1$) passing through three important points (B, C and D). It is clear that when MSF or PSF is used, all the five points from A to E



(a) MSF (b) PSF

Fig. 8: Illustrations of MSF and PSF for maintaining diversity, where dashed lines are contours.

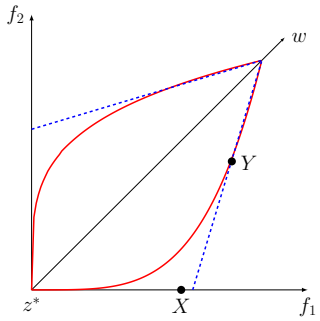


Fig. 9: An illustration where MSF (red solid) and PSF (blue dashed) induce different shapes of contours.

can survive during the replacement because no point resides in the improvement region of another point. Thus, MSF and PSF are effective and promising for diversity maintenance.

2) MSF vs PSF: The improvement region of both methods varies with α, and both degenerates to TCH when $\alpha = 0$. For $\alpha \geq 0$, however, the geometries of MSF and PSF are different. The contour lines of MSF are nonlinear whereas those of PSF are polytopes. Thus, boundary points besides the ideal point can be in an improvement region induced by PSF, but this is not the case with MSF. Fig. 9 presents a situation where a boundary point X will replace an intermediate point Y associated with the search direction w if PSF with inappropriate opening angles is used. But, this will not happen to MSF. For this reason, MSF may keep diversity better than PSF whereas PSF may have advantage in locating boundary solutions.

3) PSF vs PBI: Since PSF borrows the idea of diversity maintenance from PBI, PSF and PBI have similar contour lines, i.e., their contour lines are polytopes. However, they differ much in convergence promotion. Specifically, PBI measures convergence via d_1 values (see Fig. 1(c)). In complex problems with irregular PF shapes [20], d_1 values vary significantly. In order to balance diversity and convergence, PBI needs to carefully select the penalty factor. Otherwise, PBI is likely to obtain an incomplete approximation of the PF [35]. Unlike PBI, PSF measures convergence via TCH, which can approximate both convex and nonconvex PF geometries. Therefore, PSF may be less sensitive to different PF scenarios compared with PBI.

IV. Preliminary Experiments

As a starting point, we investigate the effectiveness of the proposed scalarizing functions in this section.

A. Experimental Settings

As our focus is mainly on the diversity aspect of scalarizing functions, test problems used for experimental validation should be able to challenge MOEAs' diversity performance. For this reason, the MOP [28] test suite is selected. This test suite has seven instances, each of which has local attractors on boundary regions of the PF. Thus, MOEAs are very easy to get trapped into these attractors if their diversity is not well maintained. A detailed description of the MOP test suite can be found in the supplementary material, where two more tri-objective instances, i.e., MOP8 and MOP9, are proposed by considering new characteristics. MOP8 places local attractors in the intermediate regions of its linear PF, whereas MOP9 increases optimization difficulties by placing local attractors only on corner regions (i.e. the intersection of the PF and coordinate axes). These added features are expected to further understanding of MOEAs' search behavior.

The proposed MSF and PSF are integrated into the MOEA/D-DE [26] framework, whose recombination operator is replaced by the adaptive operator [29] due to its reported success on MOP problems [28].

Parameters in MOEA/D-DE were set as follows. The population size N was 100 and 105 for bi- and tri-objective problems, respectively. The neighborhood size T was $T = 0.1N$ and the probability δ used to select mating neighborhood was $\delta = 0.9$. The maximal allowable number n_r of solutions to be replaced by a child solution was $n_r = 2$. Due to the difficulty of the MOP test suite, the maximum number $MaxGen$ of generations was set to $MaxGen = 5000$. The total number of independent runs was 31.

Since both MSF and PSF use Eq. (10) to assign α values, we just need to test the influence of different β values in Eq. (10). In the experimental study, β was chosen from {0, 0.05, 0.2, 1, 5, 10, 20} for MSF and {0, 1, 5, 10, 20, 100} for PSF. $\beta = 0$ is a special case where both MSF and PSF degenerate to TCH.

B. Performance Indicators

In our experimental studies, we adopt the following widely used performance indicators.

1) Inverted Generational Distance (IGD) [28]: IGD can provide reliable information on both the diversity and convergence of obtained solutions. Let P be a set of solutions uniformly sampled from the true PF, and P^* be the approximated solutions in the objective space, the indicator measures the gap between P^* and P, calculated as follows:

$$IGD(P^*, P) = \frac{\sum_{x \in P} d(x, P^*)}{|P|} \tag{13}$$

where $d(x, P^*)$ is the distance between the member x of P and the nearest member of P^*. For the calculation of IGD, P is composed of 5000 scattered points which are uniformly sampled from the true PF.

2) Averaged Hausdorff Distance (Δ_p) [37]: Δ_p is a recently developed indicator that prefers evenly spread solutions along the PF [33] and can somewhat handle the outlier tradeoff [37]. The indicator is calculated as follows:

$$\Delta_p(P^*,P)=\max\left\{\left(\frac{\sum_{x\in P} d^p(x,P^*)}{|P|}\right)^{\frac{1}{p}},\left(\frac{\sum_{x\in P^*} d^p(x,P)}{|P^*|}\right)^{\frac{1}{p}}\right\} \quad (14)$$

where $d(x,P)$ is the distance between the member x of P^* and the nearest member of P. In this paper, $p=2$ is used.

3) Hypervolume Difference (HVD): The hypervolume difference (HVD) [21] measures the gap between the hypervolume of the obtained P^* and that of the true PF:

$$HVD = HV(PF) - HV(P^*) \quad (15)$$

where $HV(S)$ is the hypervolume of a set S. The reference point for the computation of hypervolume is $(z_1+0.2, z_2+0.2, \cdots, z_M+0.2)$, where z_j is the maximum value of the jth objective of the true PF and M is the number of objectives. $HV(PF)$ can be estimated by $HV(P)$ where P remains the same in the IGD indicator.

C. Results

Figs. 10–12 plot the mean values of three indicators obtained by MSF with different β settings on some selected test problems (the results for all the test problems are presented in the supplementary material), where standard deviation is shown around the mean values. Two observations can be obtained from the figures. First, all the three indicators are roughly consistent when they are used for performance assessment. The only exception occurs on MOP7, where both IGD and Δ_p show the performance improves at first and then degrades as β increases from 0 to 20. HVD, however, shows a conflicting performance trend on MOP7. This may be because HVD prefers boundary solutions but does not necessarily favor well-diversified distribution on this particular instance. Second, for the majority of the problems, the performance is likely to be maximized when β approximately equals one. Meanwhile, it seems that a smaller β value is suitable for MOP8. This implies that, when local attractors reside in the intermediate regions of the PF, restrictions on the diversity aspect of MSF can be relaxed and MSF with a large improvement region is helpful in this situation.

On the other hand, the mean values of the three indicators obtained by PSF with different β settings on some selected test problems are displayed in Figs. 13–15. Similar observations can be obtained from these figures, and PSF works best on most of the problems when β is around 10.

The above results clearly show that both MSF and PSF can help improve decomposition-based MOEAs if the corresponding control parameter is well configured. The experiment indirectly reflects that the popular TCH method (corresponding to the case of $\beta=0$ in MSF and PSF) are not always the best choice, and enhancing diversity by nicely controlling improvement regions can lead to a clear performance improvement.

Besides, the mean HVD evolution curves obtained by MSF versus the number of generations are shown in Fig. 16 (those obtained by PSF are included in the supplementary material due to page limit). We can observe that different β values result in distinct performances. $\beta=0$ is not the best setting for MSF and PSF on the majority of the test problems in terms of final HVD values. $\beta=1$ and $\beta=10$ again help MSF and PSF yield good results, respectively. However, it seems that smaller β values are likely to converge faster, owing to relatively larger improvement regions. These observations suggest decomposition-based MOEAs may need different β (or the resulting α) values at different stages of the search. Therefore, it is plausible to adaptively adjust the value of α during the search.

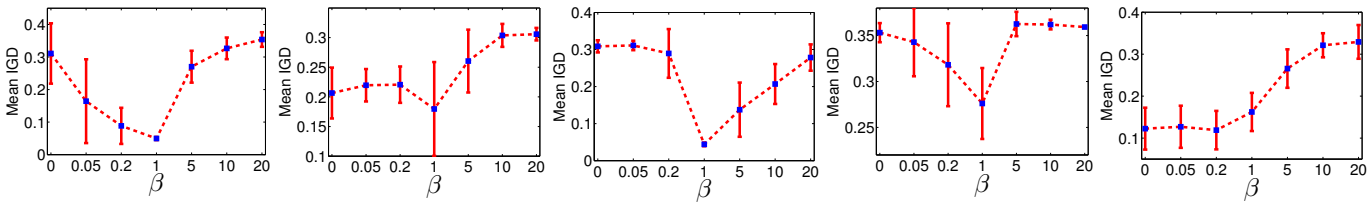
Algorithm 1: The eMOEA/D Framework

Input: stopping criterion ($MaxGen$), population size (N), neighborhood size (T), replacement size (n_r);
Output: approximated Pareto-optimal set P;

Generate a uniform spread of N weight vectors: $\{w^1, w^2, \ldots, w^N\}$ and then compute the T closest weight vectors to each weight vector by the Euclidean distance. For each w^i, set $B(i)=\{i_1,\ldots,i_T\}$ where $w^{i_1}, \ldots, w^{i_T}$ are the T closest weight vectors to w^i;
Generate an initial population $P=\{x^1,\ldots,x^N\}$ by uniformly randomly sampling from the decision space;
Initialize $ideal$ and $nadir$ points, i.e., z^* and z^{nad};
Choose a scalarizing function SF for MOEA/D;
$gen \leftarrow 1$;
while $gen \leq MaxGen$ **do**
  Update α for the selected SF according to Eq. (16);
  for $i \leftarrow 1$ **to** N **do**
    Randomly select indexes r_1 and r_2 from $B(i)$;
    Apply genetic operators on individuals x^{r_1}, x^{r_2} to produce a new solution y;
    Evaluate the objective vector of y, and update z^*;
    Find the T most suitable subproblems for y: $S=\{s_1,s_2,\ldots,s_T\}$;
    $c \leftarrow 0$;
    for $j \leftarrow 0$ **to** T **do**
      if $y \prec_{SF} x^{s_j}$ **then**
        $x^{s_j} \leftarrow y$ and $c \leftarrow c+1$;
      end
      if $c \geq n_r$ **then**
        break;
      end
    end
  end
  Update z^{nad} using P, $gen \leftarrow gen+1$;
end

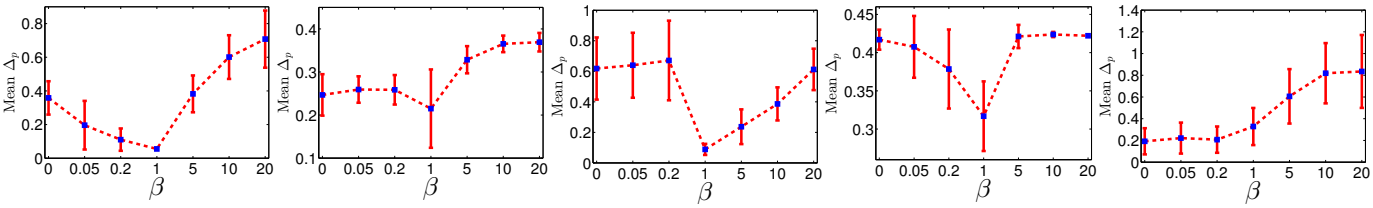
V. eMOEA/D: An Efficient MOEA/D Framework

The proposed MOEA/D variant remains almost the same as its predecessors [26], [46] except a few modifications in scalarizing methods, offspring production and solution replacement. The framework of the algorithm is depicted in Algorithm 1. First, a set of uniformly-distributed weight vectors is created,



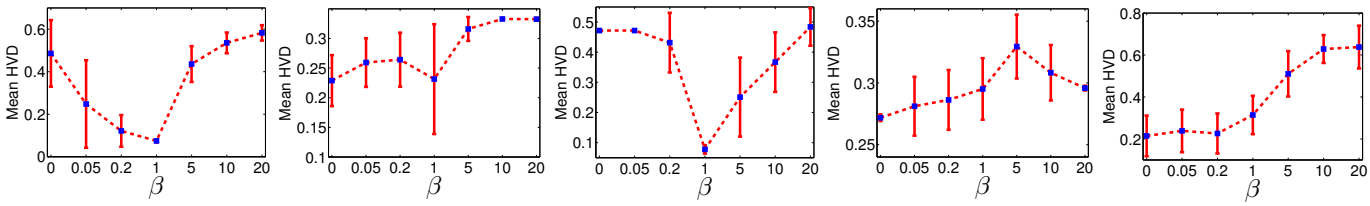
(a) MOP1 (b) MOP2 (c) MOP5 (d) MOP7 (e) MOP8

Fig. 10: Mean IGD values obtained by MSF with different β settings.



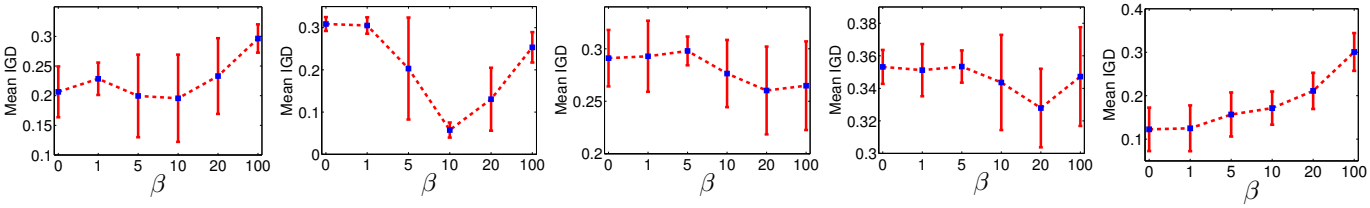
(a) MOP1 (b) MOP2 (c) MOP5 (d) MOP7 (e) MOP8

Fig. 11: Mean Δ_p values obtained by MSF with different β settings.



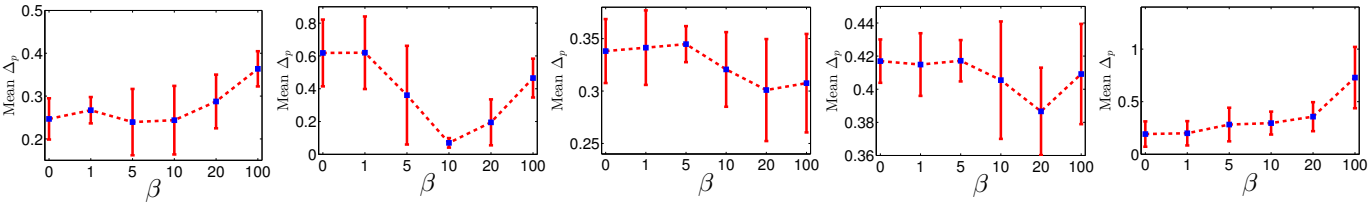
(a) MOP1 (b) MOP2 (c) MOP5 (d) MOP7 (e) MOP8

Fig. 12: Mean HVD values obtained by MSF with different β settings.



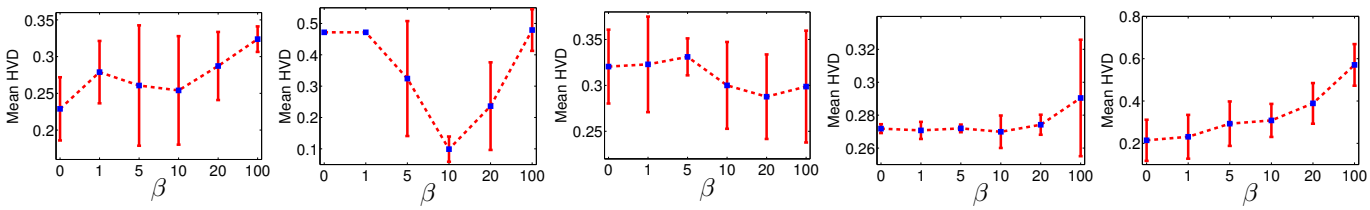
(a) MOP2 (b) MOP5 (c) MOP6 (d) MOP7 (e) MOP8

Fig. 13: Mean IGD values obtained by PSF with different β settings.



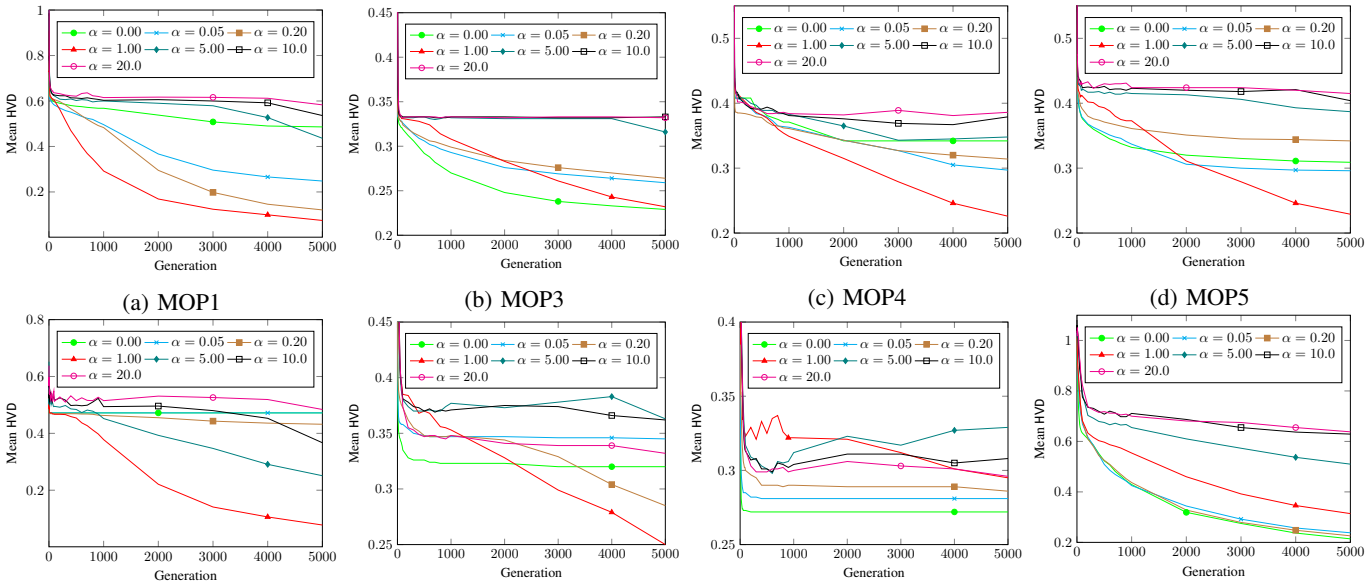
(a) MOP2 (b) MOP5 (c) MOP6 (d) MOP7 (e) MOP8

Fig. 14: Mean Δ_p values obtained by PSF with different β settings.



(a) MOP2 (b) MOP5 (c) MOP6 (d) MOP7 (e) MOP8

Fig. 15: Mean HVD values obtained by PSF with different β settings.



(a) MOP1 (b) MOP3 (c) MOP4 (d) MOP5

(e) MOP6 (f) MOP7 (g) MOP8 (h) MOP9

Fig. 16: Evolution curves of the mean HVD indicator obtained by MSF with different β settings.

and each weight vector is assigned a neighborhood containing the T closest weight vectors. Meanwhile, the ideal point (z^*) and nadir point (z^{nad}) are estimated by the minimum and maximum values of each objective in the population. z^* and z^{nad} are then used for objective normalization ($(f_i(x) - z_i^*)/(z_i^{nadir} - z_i^*)$ is adopted in this paper to normalize each objective f_i). After that, either MSF or PSF is chosen as a scalarizing function beforehand. The α value of the scalarizing function SF is generationally updated in line 7. For each subproblem i, mating parents are selected only from $B(i)$ (see line 9), which is the same as the original MOEA/D [46] but different from another popular variant MOEA/D-DE [26]. In line 10, genetic operators are applied on the selected parents to produce offspring. The offspring is evaluated in terms of the objective vector and is then used to update the ideal point. From line 12 to line 21, a new solution replacement strategy is introduced, and similar to MOEA/D-DE, we also place a restriction on the number of replacements (see lines 18-20). At the end of every generation, the approximated nadir point is updated by the whole population.

A. Adaptive Scalarizing Strategy

While the proposed scalarizing functions are helpful for maintaining population diversity, it may decrease the convergence performance. A small α value in both MSF and PSF are beneficial to convergence, but it is very likely to cause the loss of diversity. This is just the case with TCH, which struggles to recover from the loss of diversity for hard problems. Without any information about problem properties *a priori*, it is plausible to emphasize diversity at the early stage of search and then gradually emphasize convergence at the late stage. To this end, we propose an adaptive strategy to adjust the value of α (line 7 of Algorithm 1). As a result, Eq. (10) is rewritten as follows:

$$\alpha = \beta(1 - \frac{gen}{MaxGen})\Big\{m \min_{1\le i\le m}(w_i)\Big\} \qquad (16)$$

where gen is the current generation number, and $MaxGen$ is the maximum number of generations. It is clear that, for each subproblem, α is decreased linearly as the evolution proceeds and becomes zero at the end of search. This means that both MSF and PSF gradually degenerate to TCH, and in this process the improvement region for each subproblem is gradually increased, resulting in steady-state de-emphasis of diversity and speed-up of convergence simultaneously. Note that, any adaptive decreasing strategy (no matter it is linear or nonlinear) can be used as long as it can reduce α gradually. The linear strategy is adopted here because it is very simple and meets the requirement of reducing α well.

B. Reproduction Operation

Reproduction operation (lines 9–10 of Algorithm 1) includes mating pool selection and genetic recombination. In many MOEA/D variants [26], [41], a probability parameter δ is adopted to select a mating pool from either the neighborhood of solutions or the whole population. The main purpose for this is to increase population diversity. However, this induces the difficulty in tuning such an extra parameter. Since we have introduced advanced scalarizing functions that can keep diversity well, we discourage the use of the probability parameter and simply set the mating pool as the neighborhood of solutions.

When mating parents are randomly selected from the mating pool, the next step is to perform genetic operators on the mating parents to generate offspring. In this paper, we use the adaptive DE [29] as our genetic operator. The adaptive DE was also used in [28] and showed good performance for hard problems. The details of this operator are presented in the supplementary material.

C. Replacement Operation

Replacement operation (lines 12–21 of Algorithm 1) is a key step in many MOEA/D variants. It is related to what and

TABLE I: Best, median, and worst Δ_p values obtained by different algorithms on MOP problems

Prob.	MSF*	PSF*	TCH	ACD	AGR	DU	STM	M2M
MOP1	9.14E-03	9.38E-03	7.37E-02	9.50E-03	1.86E-02	1.01E-02	**8.65E-03**	1.46E-02
	9.46E-03	9.68E-03	3.97E-01‡	9.77E-03	1.99E-02‡	1.04E-02‡	9.98E-02‡	1.60E-02‡
	9.95E-03	**9.92E-03**	4.11E-01	9.95E-03	2.63E-02	1.17E-02	3.96E-01	1.75E-02
MOP2	**4.60E-03**	4.63E-03	1.89E-01	4.62E-03	7.65E-03	5.47E-03	4.72E-03	1.18E-02
	4.67E-03	4.67E-03	2.35E-01‡	4.67E-03	2.26E-02‡	5.74E-03‡	7.47E-03‡	1.36E-02‡
	4.86E-03	5.58E-03	4.12E-01	4.87E-03	2.02E-01	7.60E-03	2.76E-01	1.22E-01
MOP3	5.65E-03	5.24E-03	4.63E-01	**5.11E-03**	1.05E-02	6.52E-03	4.63E-01	1.28E-02
	6.02E-03	6.00E-03	4.93E-01‡	**5.98E-03**	3.49E-02‡	8.37E-03‡	4.63E-01‡	1.55E-02‡
	3.40E-02	4.04E-02	6.36E-01	**8.77E-03**	1.41E-01	3.76E-02	4.76E-01	2.87E-02
MOP4	**6.65E-03**	8.43E-03	3.04E-01	8.28E-03	7.08E-03	6.72E-03	6.60E-03	1.28E-02
	1.31E-02	1.31E-02	3.19E-01‡	1.33E-02	1.45E-02‡	1.32E-02	2.81E-01‡	1.55E-02‡
	2.52E-02	2.49E-02	3.74E-01	2.04E-02	9.64E-02	1.88E-02	2.90E-01	2.87E-02
MOP5	**8.58E-03**	9.11E-03	2.95E-01	1.07E-02	1.80E-02	1.54E-02	2.64E-02	1.81E-02
	3.11E-02	**1.26E-02**	6.28E-01‡	1.27E-02◇	2.32E-02‡	1.69E-02◇	1.92E-01‡	2.65E-02‡
	1.21E-01	4.14E-02	1.08E+00	1.53E-01	4.61E-02	**3.52E-02**	1.92E-01	2.12E-01
MOP6	**4.30E-02**	4.34E-02	2.83E-01	1.76E-01	8.15E-02	5.15E-02	5.69E-02	1.29E-01
	4.35E-02	4.40E-02	3.59E-01‡	2.65E-01‡	1.62E-01‡	6.10E-02‡	2.49E-01‡	1.76E-01‡
	4.69E-02	4.90E-02	3.59E-01	5.37E-01	2.66E-01	1.49E-01	2.83E-01	1.08E+00
MOP7	7.78E-02	**6.78E-02**	3.69E-01	3.25E-01	2.83E-01	8.70E-02	3.24E-01	1.89E-01
	8.30E-02	**7.21E-02**	4.21E-01‡	3.59E-01‡	3.59E-01‡	1.01E-01†	3.59E-01‡	2.46E-01‡
	1.34E-01	2.55E-01	4.21E-01	4.21E-01	4.21E-01	1.84E-01	4.21E-01	3.80E+00
MOP8	4.13E-02	**4.10E-02**	6.22E-02	1.73E-01	6.21E-02	7.01E-02	4.54E-02	2.29E-01
	4.71E-02	**4.58E-02**	1.65E-01‡	3.00E-01‡	2.10E-01‡	7.78E-02‡	5.40E-02‡	6.43E-01‡
	7.82E-02	8.74E-02	4.61E-01	4.14E-01	4.26E-01	9.30E-02	4.04E-01	1.13E+00
MOP9	**7.01E-02**	8.23E-02	3.24E-01	1.69E-01	1.64E-01	8.79E-02	3.24E-01	2.02E-01
	7.78E-02	9.17E-02	5.11E-01‡	3.67E-01‡	2.24E-01‡	9.25E-02†	3.25E-01‡	3.36E-01‡
	9.30E-02	2.40E-01	7.47E-01	9.78E-01	3.78E-01	2.13E-01	4.09E-01	1.06E+00

† and ◇ indicate MSF* and PSF* significantly outperform the corresponding algorithm, respectively.
‡ indicates both MSF* and PSF* significantly outperform the corresponding algorithm.

how subproblems can be updated. If a new solution is not suitable for subproblems that are chosen to be updated, then both population diversity and convergence can be negatively affected. To this end, various replacement strategies have been proposed [41], [49]. The main idea behind these strategies is to find the most suitable subproblem for a newly-generated individual y and then conduct replacement within the neighborhood of this subproblem. However, these strategies fail to consider that the individual y may not be good for the neighboring subproblems of the most suitable subproblem. If the individual does not improve any solution of the neighborhood of the most suitable subproblem but does improve solutions of other subproblems outside the neighborhood, it should enter the population. In other words, the replacement range should be gingerly elaborated. In this paper, the replacement range is composed of the most suitable T subproblems. It is calculated as follows. First, $g^{SF}(y|w^i, z^*)$ is computed for each $1 \leq i \leq N$. Second, all the $g^{SF}(y|w^i, z^*)$ values are sorted in the ascending order. Then, the subproblems corresponding to the first T smallest scalarizing values are regarded as the replacement range $S = \{s_1, s_2, \ldots, s_T\}$, with s_1 being the first most suitable and s_T being the T-th most suitable (line 12 of Algorithm 1).

The replacement procedure (lines 14–21 of Algorithm 1) is executed on the ordered replacement range $S = \{s_1, s_2, \ldots, s_T\}$ one by one. Like its predecessor [26], the proposed eMOEA/D framework allows at most n_r solutions to be replaced by a newly generated solution.

VI. Algorithm Comparison and Results

The experiment in this section is designed for two purposes. One is to verify the proposed eMOEA/D. The other purpose is to deeply analyze the performance of other existing decomposition-based MOEAs in multiobjective optimization.

A. Compared Algorithms and Parameter Settings

Algorithms for comparison consist of popular peer MOEAs. The MOEAs are MOEA/D with TCH [26], ACD [40], AGR [41], DU [45], STM [25], and M2M [28] schemes. For notational convenience, MSF* and PSF* denote the proposed eMOEA/D with MSF and PSF, respectively.

Key parameters in each compared algorithm remain the same as in the referenced papers. The population size, stopping criterion, and other important parameters are kept the same as in Section IV. All the algorithms use the adaptive operator [29] as the recombination operator. The key factor β in MSF* and PSF* is set to 1 and 10, respectively, based on the previous experimental study.

B. Results on MOP Problems

In this subsection, Δ_p and HVD are used as performance indicators since IGD and Δ_p have shown consistent performance assessment in the previous experiment. Tables I and II show the best, median, and worst values of Δ_p and HVD on nice MOP problems over 31 independent runs, respectively. The best values obtained by one of the ten algorithms are highlighted in bold face. The differences between the approximations are assessed by the Wilcoxon rank-sum test [42] at the 0.05 significance level, with the standard Bonferroni correction

TABLE II: Best, median, and worst HVD values obtained by different algorithms on MOP problems

Prob.	MSF*	PSF*	TCH	ACD	AGR	DU	STM	M2M
	1.39E-02	1.43E-02	7.66E-02	1.59E-02	2.73E-02	1.53E-02	**1.32E-02**	2.48E-02
MOP1	**1.43E-02**	1.46E-02	5.46E-01‡	1.61E-02‡	2.91E-02‡	1.58E-02‡	6.80E-02‡	2.70E-02‡
	1.50E-02	**1.49E-02**	5.82E-01	1.64E-02	3.56E-02	1.69E-02	5.39E-01	2.88E-02
	6.30E-03	6.48E-03	1.48E-01	6.90E-03	1.02E-02	8.39E-03	6.42E-03	1.76E-02
MOP2	**6.39E-03**	6.55E-03	2.22E-01‡	8.29E-03‡	1.82E-02‡	8.64E-03‡	6.78E-03	1.91E-02‡
	6.53E-03	8.36E-03	3.33E-01	9.51E-03	2.64E-01	1.20E-02	1.84E-01	1.36E-01
	6.08E-03	6.29E-03	2.15E-01	6.13E-03	1.03E-02	8.12E-03	2.15E-01	1.26E-02
MOP3	**6.76E-03**	6.80E-03	3.48E-01‡	7.16E-03†	1.24E-02‡	8.69E-03‡	2.15E-01‡	1.53E-02‡
	7.94E-03	1.45E-02	4.15E-01	1.03E-02	1.55E-01	3.19E-02	2.82E-01	1.88E-02
	3.45E-03	6.48E-03	2.78E-01	7.21E-03	4.35E-03	4.26E-03	**3.06E-03**	8.72E-03
MOP4	9.45E-03	**9.57E-03**	3.12E-01‡	1.33E-02‡	1.31E-02	1.02E-02	2.29E-01	1.48E-02‡
	2.10E-02	**1.54E-02**	3.48E-01	2.82E-02	1.25E-01	1.33E-02	2.45E-01	2.47E-02
	1.38E-02	1.45E-02	4.72E-01	1.70E-02	2.73E-02	2.33E-02	2.94E-02	2.92E-02
MOP5	1.63E-02	**1.60E-02**	4.72E-01‡	1.77E-02	3.12E-02‡	2.46E-02‡	3.13E-01‡	3.55E-02‡
	2.09E-02	**1.87E-02**	4.72E-01	1.88E-02	4.63E-02	3.95E-02	3.13E-01	5.45E-02
	5.15E-02	5.25E-02	2.46E-01	1.20E-01	9.28E-02	6.23E-02	5.77E-02	1.53E-01
MOP6	**5.25E-02**	5.31E-02	3.47E-01‡	2.17E-01‡	2.00E-01‡	7.66E-02‡	1.83E-01‡	1.91E-01‡
	6.32E-02	6.50E-02	3.47E-01	3.47E-01	2.91E-01	1.20E-01	2.46E-01	2.96E-01
	9.58E-02	1.08E-01	2.60E-01	1.70E-01	1.58E-01	9.67E-02	1.58E-01	1.63E-01
MOP7	**1.03E-01**	1.11E-01	2.72E-01‡	2.62E-01‡	2.65E-01‡	1.13E-01	2.55E-01‡	2.14E-01‡
	1.60E-01	2.10E-01	2.72E-01	2.98E-01	2.72E-01	1.88E-01	2.72E-01	1.20E+00
	4.44E-02	**4.38E-02**	8.65E-02	6.59E-02	6.94E-02	9.25E-02	5.62E-02	2.63E-01
MOP8	5.81E-02	**5.46E-02**	2.02E-01‡	1.16E-01‡	1.01E-01‡	1.01E-01‡	6.88E-02‡	4.29E-01‡
	9.58E-02	1.09E-01	3.81E-01	1.54E-01	1.52E-01	1.71E-01	3.15E-01	7.30E-01
	9.25E-02	**8.05E-02**	1.59E-01	1.32E-01	1.04E-01	9.71E-02	1.58E-01	2.05E-01
MOP9	**1.01E-01**	1.03E-01	6.15E-01‡	3.96E-01‡	1.72E-01‡	1.03E-01‡	1.59E-01‡	2.48E-01‡
	1.71E-01	3.25E-01	6.76E-01	6.20E-01	3.58E-01	2.23E-01	4.78E-01	4.63E-01

† and ⋄ indicate MSF* and PSF* significantly outperform the corresponding algorithm, respectively.
‡ indicates both MSF* and PSF* significantly outperform the corresponding algorithm.

[1] to deal with the problem of the higher probability of Type I errors in multiple comparisons. Signs of †, ⋄ and ‡ in the superscript form on median values indicate the significance of the proposed methods.

From the tables, we can obtain the following observations:

1) Compared with the predecessor TCH, all the other MOEA/D variants show improvements on the MOP problems in some sense. The improvements are obvious on the five bi-objective problems, as indicated by the Δ_p and HVD values. However, on the four tri-objective problems, ACD, AGR, STM, and M2M do not have any advantage over TCH, as their Δ_p and HVD values are very similar to those of TCH.
2) MSF* and PSF* significantly outperform the other algorithms on the majority of cases. On the bi-objective cases, ACD also shows comparable performance in terms of Δ_p. However, ACD degrades dramatically and performs worse than most of the algorithms on the tri-objective cases.
3) DU, aimed to improve solution replacement by considering distance to weight vectors, works well on most of the test problem, although it is not the best among all the algorithms and degrades slightly for tri-objective problems. Its good performance is probably due to the emphasis on diversity. This indirectly shows improvements on diversity management is beneficial for solving the MOP test suite.
4) Apart from dimensionality, other characteristics of the test problems also affect the compared algorithms' performance. Taking MOP6 and MOP8 for example, they have the same PF shape except that the former has local attractors in boundary regions of the PF whereas the latter has those in intermediate regions. Judging by HVD, boundary attractors are easier than intermediate ones for MSF*, PSF*, DU, and M2M, but seem more difficult for STM.
5) Additionally, if we compare the algorithms' performance on MOP7 and MOP9, we can see that most of the algorithms degrade when the number of local attractors in boundary regions decreases. This is understandable because the decrease in the number of local attractors reduces the chance of finding boundary solutions on the PF. However, it seems that such features do not influence too much the performance of the proposed methods, as the obtained HVD results vary little.

We can conclude from the above observations that MSF* and PSF* are more likely to generate good performance than the other algorithms on the MOP test suite with a wide variety of problem characteristics and optimization difficulties. This might be mainly attributed to good diversity maintenance induced by the new scalarizing functions.

Fig. 17 presents evolutionary curves of the mean IGD values obtained by some selected algorithms on two biobjective and two triobjective problems. It is clear that MSF*, PSF* and ACD are able to reduce the IGD value efficiently for the biobjective MOP1 and MOP2 as the evolution proceeds. In the case of triobjective problems like MOP6 and MOP7, only MSF* and PSF* manage to decrease the IGD value constantly during the evolution, while the other approaches seem to end up in evolutionary stagnation after 1000 generations of search.

For an inspection of the real performance of these algorithms, we also plot their PF approximations on several

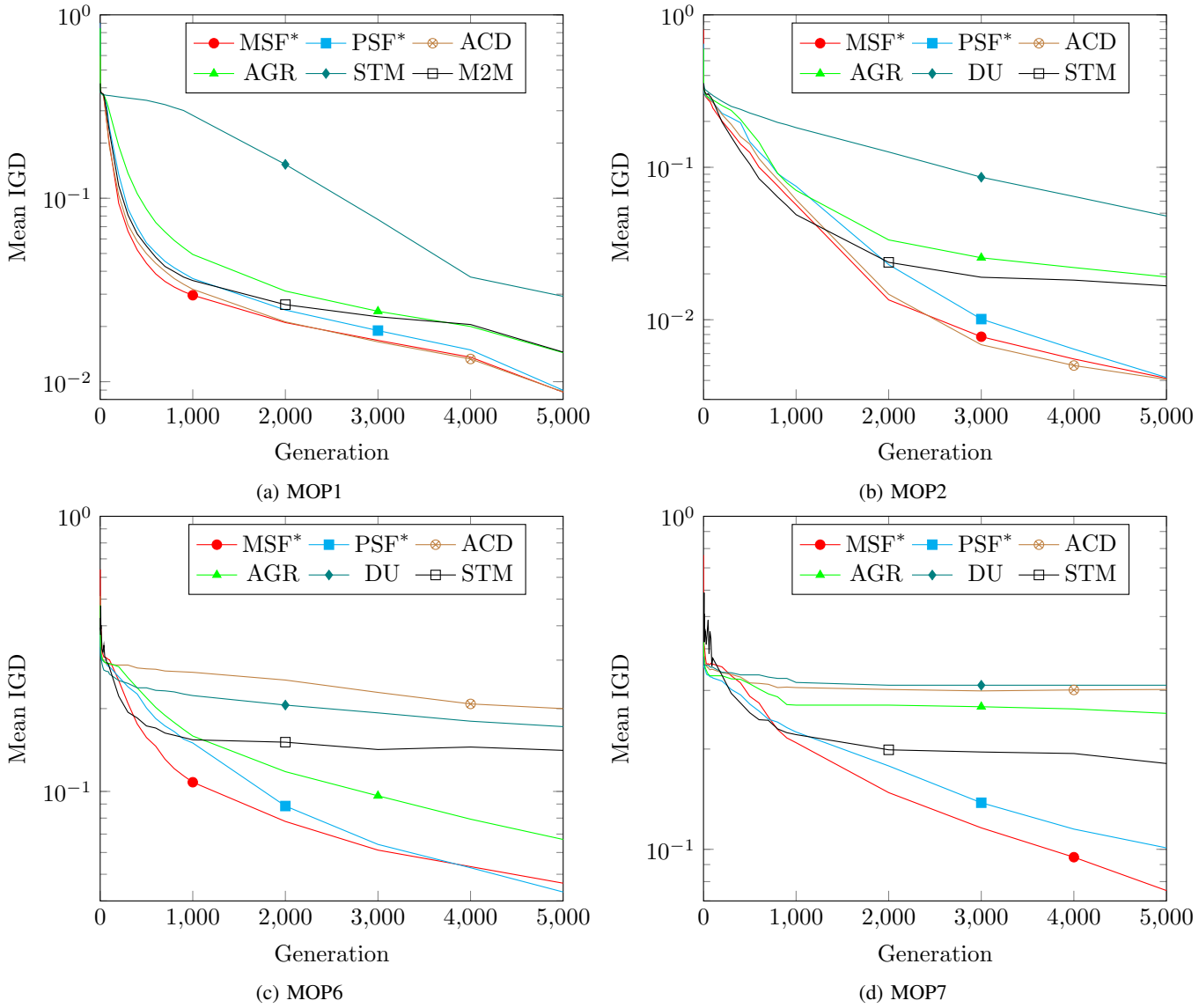


Fig. 17: Evolution curve of the mean IGD indicator obtained by six algorithms on four test problems.

selected test problems in the supplementary material. From the plots, we can observe that some algorithms (e.g., ACD, AGR, DU and M2M) converge slowly and some (e.g., TCH) cannot maintain diversity well. Also, most of them cannot work well on triobjective problems, particularly on MOP9. Nevertheless, both MSF* and PSF* show better performance compared with the other algorithms.

C. Results on UF and WFG Problems

More test problems are selected from the UF [47] and WFG [11] test suites to verify the effectiveness of the proposed algorithm. The number of variables was 30 and the maximum number of generations was equivalently set to be 3000 in UF problems, according to [25], [47]. The selected WFG problems were set to have 2 position-related variables and 10 distance-related variables in the case of two objectives, the maximum number of generations is 200.

Table III reports the HVD values obtained by ten algorithms on the selected UF and WFG test problems. It is clear to observe from the table that both MSF* and PSF* significantly outperform almost all the other compared algorithms on the UF problems considered in both biobjective and triobjective cases (UF4 and UF5 have two objectives whereas UF8 and UF9 have three objectives). There exists little difference between the ten algorithms when solving the three WFG problems as most of them obtain similar HVD values. Nevertheless, MSF* and PSF* again manage to outperform ACD, DU, and M2M on some of these WFG problems.

The experiment here also shows that the proposed eMOEA/D has great advantages over the other algorithms when solving hard-to-converge and diversity-resistant problems like the UF problems. The UF problems are hard to handle because strong nonlinear linkages between decision variables in these problems challenge dramatically EAs' diversity and convergence performance. In this situation, the adaptive scalarizing strategy in eMOEA/D can make a difference and therefore help generate promising performance. In contrast, the WFG problems are less challenging in diversity maintenance

TABLE III: Best, median, and worst HVD values obtained by different algorithms on UF and WFG problems

Prob.	MSF*	PSF*	TCH	ACD	AGR	DU	STM	M2M
UF4	**1.54E-02**	2.89E-02	1.72E-02	1.01E-01	9.23E-02	9.50E-02	8.83E-02	9.68E-02
	7.14E-02	8.56E-02	8.95E-02‡	1.07E-01‡	1.02E-01‡	1.04E-01‡	9.46E-02‡	1.01E-01‡
	1.06E-01	1.12E-01	1.69E-01	1.23E-01	1.13E-01	1.15E-01	1.09E-01	1.12E-01
UF5	**9.84E-02**	1.11E-01	1.06E-01	2.92E-01	2.69E-01	3.33E-01	2.55E-01	2.70E-01
	1.12E-01	1.19E-01	1.14E-01	4.20E-01‡	4.04E-01‡	5.02E-01‡	3.80E-01‡	3.35E-01‡
	1.23E-01	1.34E-01	1.28E-01	6.73E-01	4.75E-01	6.87E-01	5.75E-01	4.65E-01
UF8	**2.31E-02**	2.58E-02	4.01E-02	2.58E-01	2.40E-01	3.16E-01	2.49E-01	7.66E-01
	3.12E-02	3.80E-02	6.05E-02‡	4.95E-01‡	3.56E-01‡	4.87E-01‡	4.71E-01	8.54E-01‡
	2.26E-01	2.49E-01	4.97E-01	5.57E-01	5.93E-01	5.76E-01	5.27E-01	1.16E+00
UF9	2.56E-01	3.51E-01	**1.88E-01**	6.18E-01	5.98E-01	6.00E-01	4.84E-01	8.99E-01
	4.76E-01	4.91E-01	**2.16E-01**	6.87E-01‡	7.14E-01‡	7.04E-01‡	6.47E-01‡	1.07E+00‡
	5.24E-01	5.55E-01	5.87E-01	9.71E-01	1.14E+00	1.06E+00	9.90E-01	1.51E+00
WFG4	2.08E-02	2.62E-02	3.04E-02	2.50E-02	**1.67E-02**	4.35E-02	2.05E-02	2.51E-02
	3.16E-02	3.54E-02	3.77E-02	3.63E-02‡	3.05E-02	5.43E-02‡	**2.86E-02**	3.27E-02‡
	3.97E-02	4.43E-02	5.62E-02	5.42E-02	4.32E-02	7.36E-02	4.79E-02	4.68E-02
WFG5	5.98E-02	5.64E-02	6.38E-02	6.40E-02	5.53E-02	6.80E-02	**5.17E-02**	6.33E-02
	6.47E-02	6.48E-02	6.51E-02‡	6.46E-02‡	**6.41E-02**	7.49E-02‡	6.47E-02	7.08E-02‡
	6.48E-02	6.50E-02	6.62E-02	6.55E-02	**6.44E-02**	9.59E-02	6.49E-02	7.21E-02
WFG6	3.01E-02	5.12E-02	5.90E-02	2.49E-02	**9.11E-03**	6.99E-02	4.46E-02	3.17E-02
	1.64E-01	1.65E-01	1.18E-01	7.08E-02	1.64E-01	1.65E-01	**1.58E-01**	1.71E-01‡
	1.65E-01	1.66E-01	1.65E-01	1.65E-01	**1.64E-01**	1.66E-01	1.65E-01	1.73E-01

† and ◇ indicate MSF* and PSF* significantly outperform the corresponding algorithm, respectively.
‡ indicates both MSF* and PSF* significantly outperform the corresponding algorithm.

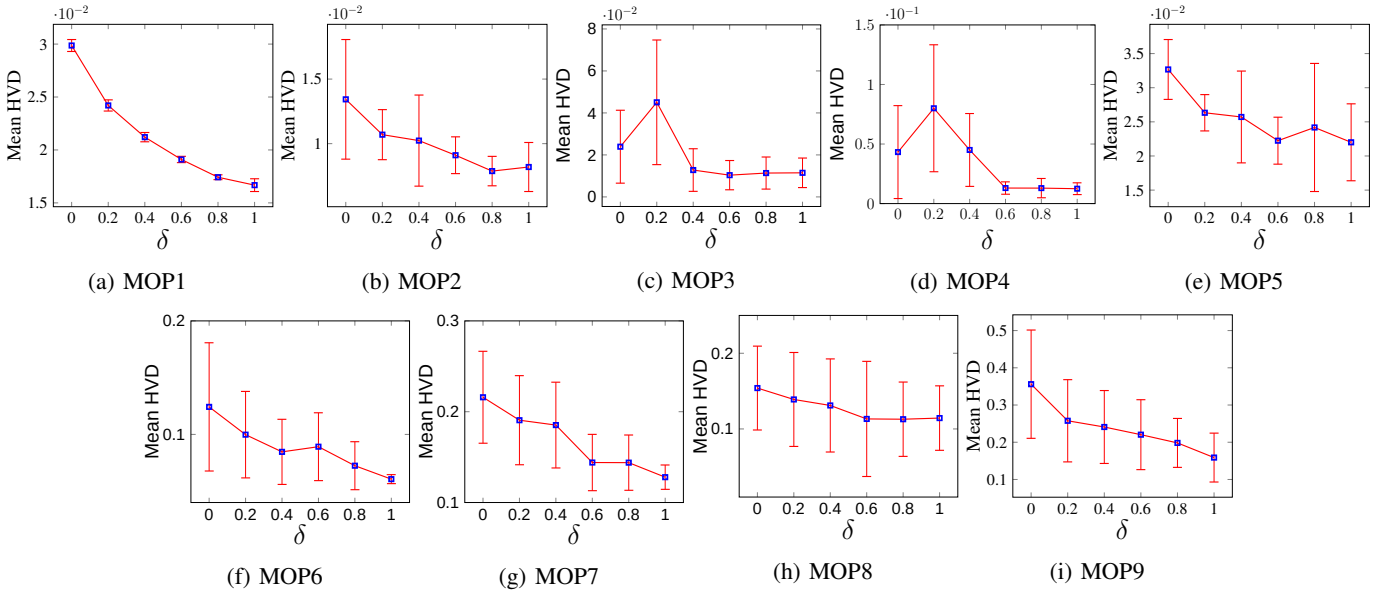


Fig. 18: Mean HVD values obtained by MSF* with different δ settings.

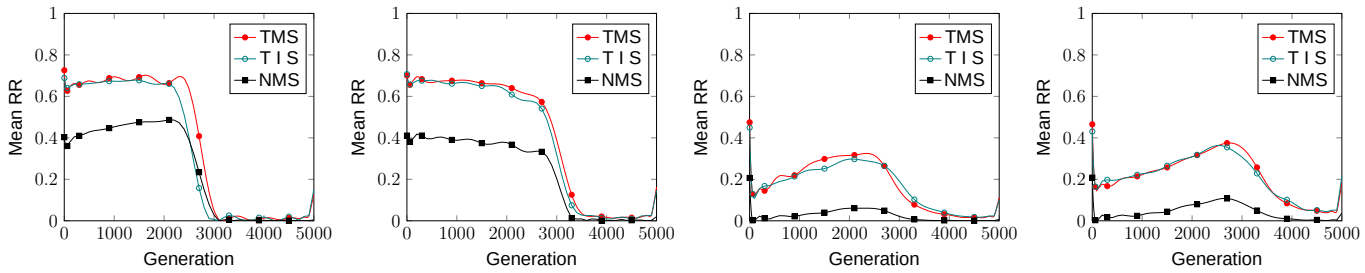
compared with the UF problems, so any algorithm with proper (not necessarily advanced) diversity management is able to solve them. Therefore, the proposed eMOEA/D has little advantage but performs comparably to the other algorithms when solving the WFG problems.

VII. Discussions

A. Influence of Mating selection

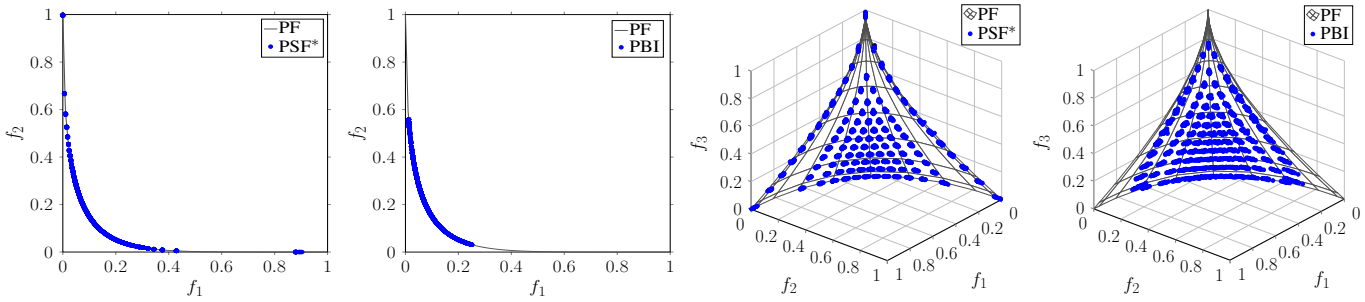
Many MOEA/D variants are developed based on the predecessor [26], and thus inevitably inherit a parameter δ that is the probability of choosing mating parents from the neighborhood of subproblems rather than the whole population. δ is of undisputed importance in the predecessor because it helps much to enhance diversity. However, most MOEA/D variants take for granted that the use of δ is always beneficial.

Here, δ from 0 to 1, with an increment of 0.2, was tested in the framework of MSF*. Fig. 18 shows the influence of δ on the obtained HVD values. It is clear that a large value of δ is roughly good for all the problems. This indicates that the higher probability of choosing subproblems' neighborhood as mating range, the better the resulting performance. This is probably because our methods have already soundly considered diversity within scalarizing functions, and in this situation using as much neighborhood mating as possible to enhance local search helps the convergence of population. Thus, in our eMOEA/D we discourage the use of δ and simply select only the neighborhood as the mating range.



(a) MOP1 (b) MOP2 (c) MOP6 (d) MOP7

Fig. 19: Evolution curves of the mean RR obtained by different replacement strategies.



(a) PSF* for F1 (b) PBI for F1 (c) PSF* for convex DTLZ2 (d) PBI for convex DTLZ2

Fig. 20: The whole PF approximations obtained by PSF* and PBI over 30 runs on two convex problems.

B. Influence of Replacement Strategies

It has been shown the performance of decomposition-based MOEAs can be significantly affected by replacement strategies [25], [41]. To achieve efficient population replacements, MOEA/D needs to find an appropriate replacement range. Most often, the replacement range is the neighborhood of the best matched subproblem [25]. In AGR [41], the best matcher is the one that has the minimal scalarizing function value, whereas in other MOEA/D variants [49], the best matcher is the one that can be improved most among all the subproblems. However, in our MOEA/D framework, the replacement range consists of the top T best matchers. In this subsection, we investigate the influence of different replacement strategies. We compare our replacement strategy (called TMS) with that of AGR (called NMS). To study the influence of the definition of matchers, we also include an replacement strategy whose replacement range is composed of the top T most improved subproblems, and this strategy is called TIS. To assess the efficiency of replacement strategies, we define the replacement rate (RR) of the population in every generation as

$$RR = \frac{N_T}{Nn_r} \tag{17}$$

where N_T is the total number of replacements that occur in the considered generation, and as stated before, N and n_r are the population size and the maximal allowable number of replacements, respectively. The larger RR, the better the replacement efficiency.

The three above-mentioned strategies have been tested in MSF* on four selected problems. Fig. 19 plots evolution curves of the mean RR value of 100 independent runs. It can be observed that replacements occur mainly at the early stage of search and the occurrence drops to near zero as the population moves close to the PF. Another observation is that NMS performs worse than TMS and TIS in terms of the RR value, and TMS is better than TIS on the two biobjective problems but is similar to TIS on the two triobjective problems. The high replacement rate of TMS helps the population evolve fast. This observation can be also used to partly explain why MSF* and PSF* perform better than AGR in the previous experiments.

C. Comparison of PSF and PBI

Since PSF and PBI have similar contour lines, it is interesting to make a comparison between them. Both PSF and PBI use our eMOEA/D framework, and accordingly the comparison objects are actually PSF* and the proposed eMOEA/D with PBI. They are investigated in two convex problems mentioned in [20]. The convex problems are chosen here because it has been increasingly recognized that irregularly-shaped problems (particularly convex ones) influence much the performance of scalarizing functions [20], [32], [35], [39]. The penalty factor of PBI was set to 5, according to [46]. Other parameter settings remained the same as in Section VI-B except the maximal number of generations was changed to 500.

The whole approximations of 31 independent runs are plotted in Fig. 20, and the corresponding Δ_p and HVD values are shown in Table IV. Both the considered indicators and the graphical plots clearly illustrate that PSF helps yield better performance than PBI. By inspecting closely the approximations in the figure, we can see that PBI favors intermediate regions of the PF and is very likely to miss boundary solutions in convex problems. This can be explained by the contour lines of

TABLE IV: Best, median and worst values of Δ_p and HVD obtained by PSF* and PBI

	Δ_p		HVD	
Prob.	PSF*	PBI	PSF*	PBI
F1	**7.53E-02**	2.93E-01	**4.51E-03**	3.34E-02
	7.57E-02	2.93E-01	**4.53E-03**	3.35E-02
	7.87E-02	2.93E-01	**4.63E-03**	3.35E-02
Convex DTLZ	**4.53E-02**	8.56E-02	**2.12E-02**	2.63E-02
	4.61E-02	9.03E-02	**2.17E-02**	2.90E-02
	4.86E-02	9.52E-02	**2.28E-02**	3.11E-02

TABLE V: Best, median and worst values of HVD obtained by different scalarizing approaches

Prob.	MSF	PSF	WS	TCH
MOP1	6.32E-02	8.08E-02	2.86E-01	7.66E-02
	7.63E-02	2.84E-01	3.76E-01	5.46E-01
	8.62E-02	5.79E-01	5.45E-01	5.82E-01
MOP2	3.12E-02	6.59E-02	3.33E-01	1.48E-01
	2.53E-01	2.51E-01	3.33E-01	2.22E-01
	3.33E-01	3.33E-01	3.33E-01	3.33E-01
MOP5	6.38E-02	7.15E-02	3.13E-01	4.72E-01
	7.36E-02	8.60E-02	3.72E-01	4.72E-01
	1.23E-01	2.20E-01	4.72E-01	4.72E-01
MOP6	1.55E-01	1.83E-01	7.34E-01	2.46E-01
	2.49E-01	3.04E-01	8.33E-01	3.47E-01
	3.25E-01	3.49E-01	8.33E-01	3.47E-01

PBI where boundary solutions likely need a very large penalty value in the PBI scalarizing function [15]. On the other hand, PSF* has a better coverage than PBI and has the potential to maintain extreme solutions and boundary solutions.

D. Further Discussions

1) Comparison with Other Scalarizing Approaches: Scalarizing functions play a fundamental role in decomposition-based EAs. Here, we would like to compare our scalarizing methods with other widely-used methods in literature. Specifically, MSF with $\alpha = 1$ and PSF with $\alpha = 10$ are compared with the WS approach and the TCH approach (PBI is excluded here because it has shown to be interior to our methods in Section VII-C). All the approaches are tested within the algorithm framework mentioned in Section IV.

Table V presents the HVD values obtained by these approaches on four MOP problems. It is clear from the table that both MSF and PSF are likely to obtain better HVD values than WS and TCH. The comparison demonstrates the promise of our scalarizing functions in helping MOEA/D to achieve high performance, which is a reason for the good performance of MSF* and PSF* in algorithm comparison shown in Section VI.

2) Investigation of Adaptive Strategies and Replacement Strategies: The proposed eMOEA/D is a combination of several strategies, i.e, new scalarizing functions, adaptive tuning of α, and a new replacement strategy. Although previous experiments have shown that the new scalarizing functions are effective and promising in decomposition-based EAs, here we want to investigate deeply the role that different strategies play in the proposed eMOEA/D.

To study each component of the two eMOEA/D instances (i.e., MSF* and PSF*), we design the following experiment. MSF with a fixed α value (i.e., $\alpha = 1$ for MSF) is compared against MSF with the proposed adaptive tuning of α (termed as "MSF+AS"), which in turn is compared against MSF*. This experimental design is also applied to PSF ($\alpha = 1$ is used). Through these two steps of comparison, one can easily see the importance of each strategy in the proposed eMOEA/D algorithm.

TABLE VI: Best, median and worst values of HVD obtained by MSF and PSF with different strategies

Prob.	MSF	MSF+AS	MSF*	PSF	PSF+AS	PSF*
MOP1	6.32E-02	1.39E-02	1.39E-02	8.08E-02	1.44E-02	1.43E-02
	7.63E-02	1.44E-02	1.43E-02	2.84E-01	1.47E-02	1.46E-02
	8.62E-02	5.55E-02	1.50E-02	5.79E-01	1.69E-02	1.49E-02
MOP2	3.12E-02	6.26E-03	6.30E-03	6.59E-02	6.51E-03	6.48E-03
	2.53E-01	6.43E-03	6.39E-03	2.51E-01	6.84E-03	6.55E-03
	3.33E-01	9.73E-02	6.53E-03	3.33E-01	2.64E-01	8.36E-03
MOP6	1.55E-01	5.46E-02	5.15E-02	1.83E-01	9.60E-02	5.25E-02
	2.49E-01	6.23E-02	5.25E-02	3.04E-01	3.36E-01	5.31E-02
	3.25E-01	1.10E-01	6.32E-02	3.49E-01	3.47E-01	6.50E-02
MOP7	2.70E-01	1.26E-01	9.58E-02	2.31E-01	1.38E-01	1.08E-01
	2.90E-01	1.72E-01	1.03E-01	2.72E-01	2.72E-01	1.11E-01
	3.78E-01	2.32E-01	1.60E-01	2.75E-01	2.72E-01	2.10E-01

Table VI presents the HVD results obtained by MSF and PSF with different strategies on several MOP problems. It can be observed that MSF+AS and PSF+AS obtain better HVD values than MSF and PSF, respectively, implying that the use of adaptive α values generates better performance than a fixed α value during the evolution. By comparing MSF+AS with MSF*, and PSF+AS with PSF*, we can clearly see that the incorporation of the new replacement strategy tends to reduce the deviation of the HVD values (the difference between the best and worst HVD values becomes smaller), which means the replacement strategy can enhance the stability of our eMOEA/D algorithm.

The experiment shows that the proposed scalarizing functions need to work collaboratively with other strategies in order to perform to the best of their ability. This is understandable because the scalarizing functions are only a method of transforming a multiobjective problem into scalar subproblems and are unable to generate multiple solutions if there is a lack of effective collaboration when solving different subproblems. The adaptive tuning of α can control the balance between diversity and convergence by changing improvement regions (it gradually increases the size of improvement regions during the evolution in eMOEA/D) induced by scalarizing functions, leading to a high level of diversity at early stages of search and fast convergence at late stages. In contrast, fixed α values emphasize diversity all the time but may affect convergence performance. Therefore, the use of adaptive tuning of α is encouraged in our work. On the other hand, the replacement strategy, as demonstrated in Section VII-B, can increase the replacement rate, which means the population evolves fast. Thus, the use of the replacement strategy will further improve the performance of eMOEA/D.

3) Potential Limitations: This research work has some potential limitations of which practitioners or interested reader may need to be aware. The first limitation is that a high number of generations, i.e., 5000, are used for the MOP test suite due to its high optimization difficulties. This setting may not be applicable in practice, particularly when limited resources like time and computational investments are available. However,

this setting makes sense when good PF approximations are the main focus. A possible way of obtaining good PF approximations with less computational resources when solving problems like the MOP ones is to use efficient reproduction operators, if any, to shorten the convergence process. Secondly, The performance of the proposed scalarizing functions or eMOEA/D may degrade with the scaling of the number of objectives, as this often happens in decomposition-based methods, and therefore may need other techniques to enhance it in the case of many-objective optimization. Thirdly, It should be acknowledged that decomposition-based methods are less robust than dominance-based ones using scalarizing functions, in particular for problems with complex (nonuniform, discrete, degenerated) Pareto fronts.

VIII. Conclusions

Decomposition-based MOEAs are an important class of methods for multiobjective optimization, and have been frequently shown to work well when proper scalarizing functions are provided. In this paper, we have proposed two new scalarizing functions which can induce controllable contours. By adjusting the size of induced improvement regions, the new scalarizing functions can easily manage population diversity. We have studied the influence of the new scalarizing functions and have demonstrated that the proposed scalarizing functions with proper improvement regions can significantly boost the performance of decomposition-based MOEAs. Additionally, we have introduced an efficient MOEA/D (i.e., eMOEA/D) framework based on the proposed scalarizing functions and some new strategies. We have compared eMOEA/D with other recently-developed approaches. The experimental results have clearly verified the effectiveness of the eMOEA/D framework.

In the current work, the proposed eMOEA/D uses a very simple adaptive strategy (i.e., linearly decreasing α in MSF and PSF) to adjust the balance between diversity and convergence at different stages of search. Despite the appealing performance, the adaptive strategy may not be the best choice because different search stages have different (not necessarily linearly-decreasing) convergence or diversity requirements. Further investigations in this direction are beneficial. In our future research, it will be also interesting to investigate the performance of the proposed scalarizing functions in many-objective optimization.

References

[1] H. Abdi, "Bonferroni and Šidák corrections for multiple comparison," in *Encyclopedia of Measurement and Statistics*, N. J. Salkind, Eds. Thousand Oaks, CA: Sage, 2007.

[2] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, 2015.

[3] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, 2016.

[4] I. Das and J. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optimiz.*, vol. 8, no. 3, pp. 631–657, 1998.

[5] K. Deb, S. Agrawwal, A. Pratap, and T. Meyarivan, " A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[6] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester: John Wiley and Sons, 2001.

[7] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[8] K. Deb and H. Jain, "An evolutionary many-Objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2014.

[9] B. Derbel, D. Brockhoff, A. Liefooghe, and S. Verel, "On the impact of multiobjective scalarizing functions," in *Proc. 8th Int. Conf. Parallel Problem Solving from Nature (PPSN XIII)*, LNCS, vol. 8672, 2014, pp. 548–558.

[10] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Inform. Sci.*, vol. 282, pp. 363–387, 2014.

[11] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 2, pp. 477–506, 2006.

[12] N. Hanse and S. Kern, "The CMA evolution strategy: A comparing review," in *Towards A New Evolutionary Computation*, vol. 192, 2006, pp. 75–102.

[13] E. Hughes, "Multiple single objective pareto sampling," in *Proc. 2003 IEEE Congr. Evol. Comput.*, 2003, pp. 2678–2684.

[14] H. Ishibuchi, N. Akedo, and Y. Nojima, "Relation between neighborhood size and MOEA/D performance on many-objective problems," in *Evol. Multi-Criterion Optim.*, Springer, 2013, pp. 459–474.

[15] H. Ishibuchi, K. Doi, and Y. Nojima, "Use of piecewise linear and nonlinear scalarizing functions in MOEA/D," in *Proc. Int. Conf. Parallel Problem Solving from Nature (PPSN VIII)*, LNCS, vol. 9921, 2016, pp. 503–513.

[16] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Evol. Multi-Criterion Optim.*, Springer, 2009, pp. 438–452.

[17] H. Ishibuchi, Y. Sakane, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. Evol. Comput.*, 2016, in press. DOI: 10.1109/TEVC.2016.2587749.

[18] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," in *Proc. 12th Ann. Conf. Genetic Evol. Comput. (GECCO 2010)*, 2010, pp. 519–526.

[19] H. Jain and K. Deb, "An evolutionary many-Objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, 2014.

[20] S. Jiang and S. Yang, "An improved multi-objective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, 2016.

[21] S. Jiang and S. Yang, "A steady-state and generational evolutionary algorithm for dynamic multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, 2017.

[22] S. Jiang and S. Yang, "A strength Pareto evolutionary algorithm based on reference direction for multi-objective and many-objective optimization," *IEEE Trans. Evol. Comput.*, 2017, in press. DOI: 10.1109/TEVC.2016.2592479.

[23] K. Li, A. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, 2014.

[24] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Inter-relationship based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, 2014.

[25] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, 2014.

[26] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2009.

[27] H. Li, Q. Zhang, and J. Deng, "Biased multiobjective optimization and decomposition algorithm," *IEEE Trans. Cybern.*, vol. 47, no. 1, pp. 52–66, 2017.

[28] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, 2014.

[29] H. Liu and X. Li, "The multiobjective evolutionary algorithm based on determined weight and sub-regional search," in *Proc. Congr. Evol. Comput.*, vol. 15, 2009, pp. 361–384.

[30] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proc. Evol. Multi-Criterion Optim.*, vol. 1993, LNCS, 2001, pp. 82–95.

[31] P. Sebah and X. Gourdon, "Introduction to the Gamma Functions," 2002.

[32] T. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[33] G. Rudolph, O. Schtze, C. Grimme, C. Dominguez-Medina, and H. Trautmann, "Optimal averaged Hausdorff archives for bi-objective problems: theoretical and numerical results," *Comput. Optim. Appl.*, vol. 64, no. 2, pp. 589–618, 2016.

[34] R. Saborido, A. B. Ruiz, and M. Luque, "Gobal WASF-GA: An evolutionary algorithm in multiobjective optimization to approximate the whole Pareto optimal front," *Evol. Comput.*, 2016, in press. DOI: 10.1162/EVCO_a_00175.

[35] H. Sato, "Analysis of inverted PBI and comparison with other scalarizing functions in decomposition based MOEAs," *J. Heuristics*, vol. 21, no. 6, pp. 819–849, 2015.

[36] H. Sato, "Chain-reaction solution update in MOEA/D and its effects on multi- and many-objective optimization," *Soft Comput.*, vol. 20, no. 10, pp. 3803–3820, 2016.

[37] O. Schütze, X. Esquivel, A. Lara, and C. A. Coello Coello, "Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, no. 16, no. 4, pp. 504–522, 2012.

[38] Y. Tan, Y. Jiao, H. Li, and X. Wang, "MOEA/D+uniform design: A new version of MOEA/D for optimization problems with many objectives," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1648–1660, 2013.

[39] R. Wang, Q. Zhang, and T. Zhang, "Decomposition based algorithms using Pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821–837, 2016.

[40] L. Wang, Q. Zhang, A. Zhou, M. Gong, and, L. Jiao, "Constrained subproblems in decomposition based multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, vol. 3, pp. 475–480, 2015.

[41] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and, L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 474–486, 2016.

[42] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.

[43] S. Yang, S. Jiang, and Y. Jiang, "Improving the multiobjective evolutionary algorithm based on decomposition," *Soft Computing*, 2016, in press. DOI: DOI10.1007/s00500-016-2076-3.

[44] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, 2015.

[45] Y. Yuan, H. Xu, B. Wang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, 2015.

[46] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[47] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," Work. Rep. CES-491, School of CS and EE, University of Essex, Colchester, UK, 2009.

[48] S. Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442–446, 2012.

[49] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, 2016.

[50] E. Zitzler and S. Kunzli, "Indicator-based selection in multiobjective search," in *Proc. 8th Int. Conf. Parallel Problem Solving from Nature (PPSN VIII)*, LNCS, vol. 3242, 2004, pp. 832–842.

[51] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Des. Optimiz. Contr. Applicat. Indust. Problems*, EUROGEN, vol. 3242, no. 103, 2002, pp. 95-100.

Shouyong Jiang received the B.Sc. degree in information and computation science and the M.Sc. degree in control theory and control engineering from Northeastern University, Shenyang, China in 2011 and 2013, respectively. He received the PhD degree in Computer Science from De Montfort University, U.K., in 2017.

He is currently a postdoc research assistant within the School of Computing Science, Newcastle University, Newcastle upon Tyne, U.K. His current research interests include computational modelling, network analysis, evolutionary computation, and dynamic optimization.

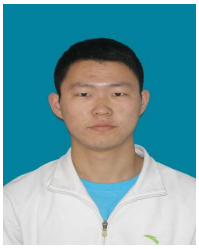
Shengxiang Yang (M'00–SM'14) received the B.Sc. and M.Sc. degrees in automatic control and the Ph.D. degree in systems engineering from Northeastern University, Shenyang, China in 1993, 1996, and 1999, respectively.

He is currently a Professor in Computational Intelligence and Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 230 publications. His current research interests include evolutionary computation, swarm intelligence, computational intelligence in dynamic and uncertain environments, artificial neural networks for scheduling, and relevant real-world applications.

Prof. Yang is the Chair of the Task Force on Evolutionary Computation in Dynamic and Uncertain Environments, under the Evolutionary Computation Technical Committee of the IEEE Computational Intelligence Society and the Founding Chair of the Task Force on Intelligent Network Systems, under the Intelligent Systems Applications Technical Committee of the IEEE Computational Intelligence Society. He serves as an Associate Editor/Editorial Board Member of 8 international journals, such as the *IEEE Transactions on Cybernetics*, *Information Sciences*, *Evolutionary Computation*, *Neurocomputing*, and *Soft Computing*.

Yong Wang (M'08) received the B.S. degree in automation from the Wuhan Institute of Technology, Wuhan, China, in 2003, and the M.S. degree in pattern recognition and intelligent systems and the Ph.D. degree in control science and engineering both from the Central South University (CSU), Changsha, China, in 2006 and 2011, respectively. He is currently an Associate Professor with the School of Information Science and Engineering, CSU. His current research interests include the theory, algorithm design, and applications of computational intelligence.

Dr. Wang is a member of the IEEE CIS Task Force on Nature-Inspired Constrained Optimization and the IEEE CIS Task Force on Differential Evolution. He was awarded the Hong Kong Scholar by the Mainland - Hong Kong Joint Postdoctoral Fellows Program, China, in 2013, the Excellent Doctoral Dissertation by Hunan Province, China, in 2013, the New Century Excellent Talents in University by the Ministry of Education, China, in 2013, the 2015 IEEE Computational Intelligence Society Outstanding PhD Dissertation Award, the Hunan Provincial Natural Science Fund for Distinguished Young Scholars, in 2016, and the EU Horizon 2020 Marie Sklodowska-Curie Fellowship, in 2016. He is currently serving as an associate editor for the journal Swarm and Evolutionary Computation.



Xiaobin Liu received the B.Sc. degree in automation from Huaqiao University, Fujian, China, in 2011, and the M.Sc. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2013.

He is currently pursuing the Ph.D. degree in the College of Information Science and Engineering, Northeastern University, Shenyang, China. His current research interests include evolutionary computation, multi-objective optimization, and many-objective optimization.

Scalarizing Functions in Decomposition-based Multiobjective Evolutionary Algorithms —Supplementary Material

Shouyong Jiang, Shengxiang Yang, *Senior Member, IEEE*, Yong Wang, *Member, IEEE*, and Xiaobin Liu

This is the supplementary material to the paper entitled "Scalarizing Functions in Decomposition-based Multiobjective Evolutionary Algorithms", written by Shouyong Jiang, Shengxiang Yang, Yong Wang and Xiaobin Liu. This material provides a detailed description of theoretical analysis of the scalarizing functions, the recombination operator and the MOP test suite used in the paper. Additional experimental results are also presented in this material.

IX. M-dimensional Improvement Region of MSF

Theorem 2. *For m ($m \geq 2$) objectives, the maximum size of the improvement region enclosed by $g^{msf}(x|w,z^*) = c$ ($c \geq 0$), denoted $\Delta_m(c)$, is equal to*

$$\Delta_m(c)=\begin{cases}\frac{w_1 w_2 c^2}{2\alpha+1}, & if\ m=2,\\ \frac{\left(\frac{c^2\pi(m-1)}{m}\right)^{\frac{m-1}{2}}\left(\prod_{i=1}^{m} w_i\right)(m-1)!}{\Gamma(\frac{m-1}{2}+1)(m(m\alpha+1)\cdots(m\alpha+(m-1)))}, & otherwise.\end{cases} \tag{1}$$

Proof: It is easy to see from Theorem 1 that, when $m = 2$, $\Delta_m(c)= \frac{w_u w_v c^2}{2\alpha+1}$ holds.

The following is devoted to proving the theorem in the case of $m > 2$. For notational convenience, let $\bar{f}_k = \bar{f_m} = \tilde{f}_k/w_k$, $k \in \{1,\ldots,m\}$, and let u and v again be the indices that maximize and minimize $\tilde{f}_i/w_i$ ($1 \leq i \leq m$), respectively.

For $m > 2$, $g^{msf}(x|w,z^*) = c$ has two apexes, i.e., (0,…, 0) and (c,…,c). Its maximum improvement region is bounded by a closed hypersurface generated by rotating the following $(m-1)$ dimensional curve C_m about the hyper-line L: $\frac{\bar{f}_1}{1} = \frac{\bar{f}_2}{1} = \cdots = \frac{\bar{f_m}}{1}$.

$$C_m : \begin{cases} \bar{f_u}^{\alpha+1} = c\bar{f_v}^{\alpha} \\ \bar{f}_k = fv, k \neq u, k \neq v, k \in \{1,\ldots,m\} \end{cases} \tag{2}$$

where C_m is actually the intersection of the above $(m-1)$ hypersurfaces. Therefore, the hyper-line L is an axis of symmetry of the maximum improve region.

Shouyong Jiang and Shengxiang Yang are with the Centre for Computational Intelligence (CCI), School of Computer Science and Informatics, De Montfort Univesity, The Gateway, Leicester LE1 9BH, U.K. (email: shouyong.jiang@email.dmu.ac.uk, syang@dmu.ac.uk).

Yong Wang is with School of Information Science and Engineering, Central South University, Changsha 410083, China (email: ywang@csu.edu.cn).

Xiaobin Liu is with School of Information Science and Engineering, Northeastern University, Shenyang 110819, China.

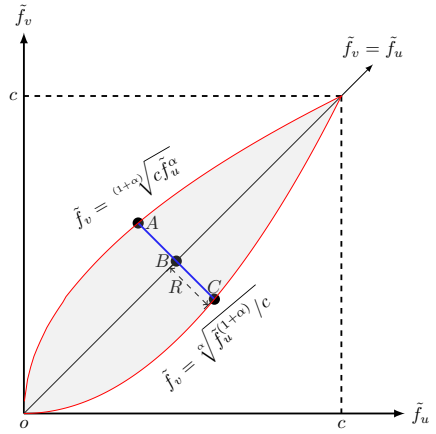


Fig. 21: The improvement region (shaded area) of MSF.

Thus, the maximum improvement region is a solid of revolution, the volume ($\bar{\Delta_m}(c)$) of which can be calculated by

$$\bar{\Delta_m}(c) = \int_0^{\sqrt{m}c} V(s)ds, \tag{3}$$

where $V(s)$ is the cross-sectional area of the solid, which is the volume of a $(n-1)$ dimensional hypersphere, and $ds = \sqrt{m}d\bar{f}_k$, $k \in \{1,\ldots,m\}$. To calculate $V(s)$, we need the radius of the hypersphere, denoted R.

R is actually the distance of a point $P = (\bar{f}_2,\ldots,\bar{f_m})^T$ from the curve C_m to the line L (or its directional vector $w = (1,\ldots,1)^T$), and it can be calculated by

$$R=\|OP\|\sqrt{1-\left(\frac{<OP,w>}{\|OP\|\|w\|}\right)^2} = \sqrt{\sum_{k=1}^{m}\bar{f}_k^2-(\sum_{i=1}^{m}\bar{f}_i/m)^2}. \tag{4}$$

Fig. 21 illustrates an example of the radius R in the case of $m = 2$.

Eq. (4) can be simplified by using Eq. (2), as follows:

$$R = \sqrt{\frac{m-1}{m}}(\bar{f}_u - \bar{f}_v). \tag{5}$$

Then, the volume of a $(m-1)$ dimensional hypersphere of radius R, according to [5], is

$$V(s) = \frac{\pi^{\frac{m-1}{2}}R^{(m-1)}}{\Gamma(\frac{m-1}{2}+1)} \tag{6}$$

where the gamma function is a generalization of the factorial function.

Now, combining Eqs. (2), (3), (5), and (6), we have

$$\begin{aligned}
&\bar{\Delta_m}(c) \qquad (7)\\
&= \int_0^{\sqrt{m}c} V(s)ds\\
&= \int_0^{c} \left[\frac{\pi^{\frac{m-1}{2}} \left(\sqrt{\frac{m-1}{m}}(\bar{f_u}-\bar{f_v})\right)^{(m-1)}}{\Gamma(\frac{m-1}{2}+1)} \right] d\bar{f_u}\\
&= \frac{\left(\frac{\pi(m-1)}{m}\right)^{\frac{(m-1)}{2}}}{\Gamma(\frac{m-1}{2}+1)} \int_0^{c} \left(\bar{f_u} - \frac{1}{\sqrt[\alpha]{c}} \bar{f_u}^{1+1/\alpha}\right)^{m-1} d\bar{f_u}\\
&= \frac{\left(\frac{c^2\pi(m-1)}{m}\right)^{\frac{(m-1)}{2}}}{\Gamma(\frac{m-1}{2}+1)} \int_0^{1} t^{m-1}(1-t^{1/\alpha})^{m-1} dt.
\end{aligned}$$

Since $\int_0^1 t^{(p-1)}(1-t^{1/\alpha})^q dt = \alpha \int_0^1 t^{p\alpha-1}(1-t)^q dt$, let us define a function

$$I(p,q,k) = \int_0^1 t^{p\alpha-1+k}(1-t)^q dt. \qquad (8)$$

Then, we have

$$\begin{aligned}
&I(p,q,k) \qquad (9)\\
&= \int_0^1 t^{p\alpha-1+k}(1-t)^q dt\\
&= \frac{1}{p\alpha+k} \int_0^1 t^{p\alpha+k}(1-t)^{q-1} dt\\
&= \frac{1}{p\alpha+k} I(p,q-1,k+1)\\
&= \frac{q!}{(p\alpha+k)(p\alpha+k+1)\cdots(p\alpha+k+q-1)} I(p,0,k+q)\\
&= \frac{q!}{(p\alpha+k)(p\alpha+k+1)\cdots(p\alpha+k+q-1)(p\alpha+k+q)}
\end{aligned}$$

Therefore, $I(m,m-1,0) = \frac{(m-1)!}{m\alpha(m\alpha+1)\cdots(m\alpha+(m-1))}$, and $\bar{\Delta_m}(c)$ in Eq. (7) will be

$$\begin{aligned}
\bar{\Delta_m}(c) &= \alpha I(m,m-1,0) \qquad (10)\\
&= \frac{\left(\frac{c^2\pi(m-1)}{m}\right)^{\frac{(m-1)}{2}}}{\Gamma(\frac{m-1}{2}+1)} \frac{(m-1)!}{m(m\alpha+1)\cdots(m\alpha+(m-1))}.
\end{aligned}$$

However, $\bar{\Delta_m}(c)$ is derived from a translated coordinate system instead of the original one. Actually, in the original coordinate system, $V(s)$ in Eq. (6) should be calculated on a solid hyperellipsoid, and $d\bar{f_u}$ in Eq. (7) should be $w_u(d\tilde{f_u})$, thus the maximum volume of the improvement region is given by

$$\begin{aligned}
\Delta_m(c) &= \left(\prod_{i=1}^{m} w_i\right) \bar{\Delta_m}(c)\\
&= \frac{\left(\frac{c^2\pi(m-1)}{m}\right)^{\frac{m-1}{2}} \left(\prod_{i=1}^{m} w_i\right)(m-1)!}{\Gamma(\frac{m-1}{2}+1)\,(m(m\alpha+1)\cdots(m\alpha+(m-1)))}.
\end{aligned} \qquad (11)$$

The proof is complete. ■

X. Theoretical Results on MSF and PSF

Theorem 3. *Let $x^* \in \Omega_x$ be Pareto optimal. Then there exists a weight vector w (all components are nonegative) such that x^* is a solution of the problem of MSF with $\alpha \geq 0$.*

Proof: Let $x^* \in \Omega_x$ be Pareto optimal. Let us assume that there does not exist a weight vector w such that x^* is a solution of the problem of MSF. We know that $f_i(x) \geq z_i^*$ for all $i = 1,\ldots,M$ and for all $x \in \Omega_x$. Now we choose $w_i = (f_i(x^*) - z_i^*)/\xi$ for all $i = 1,\ldots,M$, where $\xi \geq 0$ is some normalizing factor [4].

If x^* is not a solution of the problem of MSF, there exists another point $x^o \in \Omega_x$ that is a solution of the problem of MSF, meaning that $g^{msf}(x^o|w,z^*) < g^{msf}(x^*|w,z^*)$. That is,

$$\begin{aligned}
\frac{\left[\max\limits_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x^o)-z_i^*|\right)\right]^{1+\alpha}}{\left[\min\limits_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x^o)-z_i^*|\right)\right]^{\alpha}} &< \frac{\left[\max\limits_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x^*)-z_i^*|\right)\right]^{1+\alpha}}{\left[\min\limits_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x^*)-z_i^*|\right)\right]^{\alpha}}\\
&= \frac{\left[\max\limits_{1\leq i\leq m}\left(\frac{\xi|f_i(x*)-z_i^*|}{|f_i(x^*)-z_i^*|}\right)\right]^{1+\alpha}}{\left[\min\limits_{1\leq i\leq m}\left(\frac{\xi|f_i(x^*)-z_i^*|}{|f_i(x^*)-z_i^*|}\right)\right]^{\alpha}}\\
&= \xi.
\end{aligned}$$

Thus, we have $\left[\max\limits_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x^o)-z_i^*|\right)\right]^{1+\alpha} < \xi\left[\min\limits_{1\leq i\leq m}\left(\frac{1}{w_i}|f_i(x^o)-z_i^*|\right)\right]^{\alpha}$ for all $i = 1,\ldots,M$. This means that

$$\left[\max_{1\leq i\leq m}\left(\frac{\xi|f_i(x^o)-z_i^*|}{|f_i(x^*)-z_i^*|}\right)\right]^{1+\alpha} < \xi\left[\min_{1\leq i\leq m}\left(\frac{\xi|f_i(x^o)-z_i^*|}{|f_i(x^*)-z_i^*|}\right)\right]^{\alpha}$$

and after simplifying the expression we have

$$\left[\max_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right)\right]^{1+\alpha} < \left[\min_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right)\right]^{\alpha}. \qquad (12)$$

Eq. (12) holds if and only if $f_i(x^o) < f_i(x^*)$ for $i = 1,\ldots,M$, and this is because if x^o and x^* are nondominated, then $\max\limits_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right) > 1$ and $\min\limits_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right) < 1$, leading to

$$\left[\max_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right)\right]^{1+\alpha} > \left[\min_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right)\right]^{\alpha}$$

in the case of $\alpha \geq 0$. This is contradictory to Eq. (12).

If x^o is dominated by x^*, then we have

$$\max_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right) > \min_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right) > 1.$$

This results in

$$\left[\max_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right)\right]^{1+\alpha} > \left[\min_{1\leq i\leq m}\left(\frac{f_i(x^o)-z_i^*}{f_i(x^*)-z_i^*}\right)\right]^{\alpha}$$

in the case of $\alpha \geq 0$. This is also contradictory to Eq. (12).

However, the conclusion that Eq. (12) holds if and only if $f_i(x^o) < f_i(x^*)$ for $i = 1,\ldots,M$ is also contradictory to the Pareto optimality of x^*, which completes the proof. ■

Theorem 4. *Let $x^* \in \Omega_x$ be Pareto optimal. Then there exists a weight vector w (all components are nonegative) such that x^* is a solution of the problem of PSF with $\alpha \geq 0$.*

Proof: Let $x^* \in \Omega_x$ be Pareto optimal. Let us assume that there does not exist a weight vector w such that x^* is a solution of the problem of PSF. We know that $f_i(x) \geq z_i^*$ for all $i = 1, \ldots, M$ and for all $x \in \Omega_x$. Now we choose $w_i = (f_i(x^*) - z_i^*)/\xi$ for all $i = 1, \ldots, M$, where $\xi \geq 0$ is some normalizing factor.

If x^* is not a solution of the problem of PSF, there exists another point $x^o \in \Omega_x$ that is a solution of the problem of PSF, meaning that $g^{psf}(x^o|w, z^*) < g^{psf}(x^*|w, z^*)$. That is,

$$\begin{aligned}
&\max_{1\leq i\leq m}\Big(\frac{|f_i(x^o)-z_i^*|}{w_i}\Big) + \alpha d(x^o, w) \\
&< \max_{1\leq i\leq m}\Big(\frac{|f_i(x^*)-z_i^*|}{w_i}\Big) + \alpha d(x^*, w) \\
&= \max_{1\leq i\leq m}\Big(\frac{|\xi w_i|}{w_i}\Big) + \alpha\frac{\sqrt{\|\xi w\|^2\|w\|^2 - \|(\xi w)^T w\|^2}}{\|w\|} \\
&= \xi
\end{aligned}$$

where $d(x, w)$ is the perpendicular distance from a point x to a vector w, which is defined as

$$d(x, w) = \frac{\sqrt{\|f(x) - z^*\|^2\|w\|^2 - \|(f(x) - z^*)^T w\|^2}}{\|w\|}.$$

Thus, we have $\max_{1\leq i\leq m}\Big(\frac{|f_i(x^o)-z_i^*|}{w_i}\Big) < \max_{1\leq i\leq m}\Big(\frac{|f_i(x^o)-z_i^*|}{w_i}\Big) + \alpha d(x^o, w) < \xi$ for all $i = 1, \ldots, M$. This means that

$$\max_{1\leq i\leq m}\Big(\frac{\xi(f_i(x^o) - z_i^*)}{(f_i(x*) - z^*)}\Big) = \max_{1\leq i\leq m}\Big(\frac{|f_i(x^o)-z_i^*|}{w_i}\Big) < \xi$$

and after simplifying the expression we have

$$f_i(x^o) < f(x^*) \text{ for all } i = 1, \ldots, M. \tag{13}$$

Here, we have a contradiction with the Pareto optimality of x^*, which completes the proof. ■

XI. Liu and Li's Recombination Operator (LLX)

A. Crossover

Suppose x^i and x^j are mating parents, a child y can be generated by

$$y = x^i + rc(x^i - x^j), \tag{14}$$

with

$$rc = (2rnd - 1)(1 - rnd^{-(1-g/Mg)^{0.7}}), \tag{15}$$

where rnd is a random number in [0,1]. g and Mg are the current generation and the maximum number of generations, respectively. After that, y is repaired by

$$y_k = \begin{cases} y_k, & \text{if } a_k \leq \bar{y_k} \leq b_k \\ a_k + \frac{rnd}{2}(y_k - a_k), & \text{if } \bar{y_k} < a_k \\ b_k + \frac{rnd}{2}(b_k - a_k), & \text{if } \bar{y_k} > b_k \end{cases} \tag{16}$$

where y_k is the k-th component of y, and a_k and b_k are the lower and upper bounds of the k-th variable.

B. Mutation

Each component of y is mutated by

$$y_k = y_k + rm(b_k - a_k) \tag{17}$$

with

$$rm = 0.5(rnd - 0.5)(1 - rnd^{-(1-g/Mg)^{0.7}}) \tag{18}$$

After mutation, y is repaired by Eq. (16).

XII. MOP Test Suite

Table VII presents the formulation of the MOP test suite, including two new tri-objective problems (i.e., MOP8 and MOP9). The number of decision variables is $n = 10$, and the search space of these problems are $[0,1]^n$.

XIII. Empirical Results of The MSF and PSF Methods

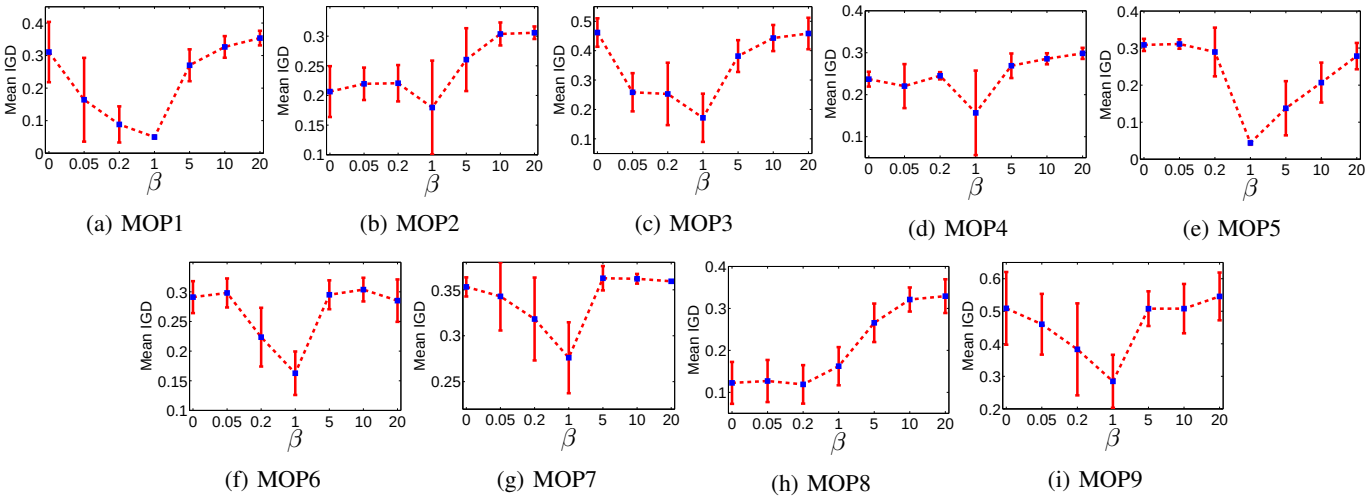
Figs. 22–24 plot the mean values of three metrics obtained by MSF with different β settings on all the MOP test problems, where standard deviations are shown around the mean values. Two observations can be obtained from the figures. First, all the three metrics are roughly consistent when they are used for performance assessment. The only exception occurs on MOP7, where both IGD and Δ_p show that the performance improves at first and then degrades as β increases from 0 to 20. HVD, however, shows a conflicting performance trend on MOP7. This may be because HVD prefers boundary solutions but does not necessarily favor well-diversified distribution on this particular instance. Second, for the majority of the problems, the performance is likely to be maximized when β approximately equals one. Meanwhile, it seems that a smaller β value is suitable for MOP8. This implies that, when local attractors reside in the intermediate regions of the PF, restrictions on the diversity aspect of MSF can be relaxed and MSF with a large improvement region is helpful in this situation.

On the other hand, the mean values of the three metrics obtained by PSF with different β settings on some selected test problems are displayed in Figs. 25–27. Similar observations can be obtained from these figures, and PSF works best on most of the problems when β is around 10.

Figs. 28 show the mean HVD evolution curves obtained by PSF versus the number of generations. We can observe that different β values result in distinct performances. $\beta = 0$ is not the best setting for PSF on the majority of the test problems in terms of final HVD values. $\beta = 10$ helps PSF yield good results. However, it seems that smaller β values are likely to converge faster, owing to relatively larger improvement regions. These observations suggest decomposition-based MOEAs may need different β (or the resulting α) values at different stages of search. Therefore, it is plausible to adaptively adjust the value of α during the search.

TABLE VII: MOP Test Suite

Instance	Description	PS/PF	Remarks
MOP1	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1})$ $g(x) = 2\sin(\pi x_1)\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i\rvert^{0.6})$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$	PS: $\{x \in R^{(n-1)} \lvert x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \lvert f_2 = 1-\sqrt{f_1}, f_1 \in [0,1]\}$	Two extremal attractors Convex PF
MOP2	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-x_1^2)$ $g(x) = 10\sin(\pi x_1)\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+\exp(5\lvert t_i\rvert)}$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$	PS: $\{x \in R^{(n-1)} \lvert x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \lvert f_2 = 1-f_1^2, f_1 \in [0,1]\}$	Two extremal attractors Concave PF
MOP3	$f_1(x) = (1+g(x))\cos(0.5\pi x_1)$ $f_2(x) = (1+g(x))\sin(0.5\pi x_1)$ $g(x) = 10\sin(\pi x_1)\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+\exp(5\lvert t_i\rvert)}$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$	PS: $\{x \in R^{(n-1)} \lvert x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \lvert f_2 = \sqrt{1-f_1^2}, f_1 \in [0,1]\}$	Two extremal attractors Concave PF
MOP4	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1}\cos^2(2\pi x_1))$ $g(x) = 10\sin(\pi x_1)\sum_{i=2}^{n}\frac{\lvert t_i\rvert}{1+\exp(5\lvert t_i\rvert)}$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$	PS: $\{x \in R^{(n-1)} \lvert x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \lvert f_2 = 1-\sqrt{f_1}, f_1 \in [0,1]\}$	Two extremal attractors Disconnected PF
MOP5	$f_1(x) = (1+g(x))x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1})$ $g(x) = 2\cos(\pi x_1)\sum_{i=2}^{n}(-0.9t_i^2 + \lvert t_i\rvert^{0.6})$ $t_i = x_i - \sin(0.5\pi x_1),\ i = 2,\ldots,n$	PS: $\{x \in R^{(n-1)} \lvert x_i = \sin(0.5\pi x_1), 2 \le i \le n, x_1 \in [0,1]\}$ PF: $\{(f_1,f_2) \lvert f_2 = 1-\sqrt{f_1}, f_1 \in [0,1]\}$	One intermediate attractor Convex PF
MOP6	$f_1(x) = (1+g(x))x_1x_2$ $f_2(x) = (1+g(x))x_1(1-x_2)$ $f_3(x) = (1+g(x))(1-x_1)$ $g(x) = 2\sin(\pi x_1)\sum_{i=3}^{n}(-0.9t_i^2 + \lvert t_i\rvert^{0.6})$ $t_i = x_i - x_1x_2,\ i = 3,\ldots,n$	PS: $\{x \in R^{(n-2)} \lvert x_i = x_1x_2, 2 \le i \le n, x_j \in [0,1], j = 1,2\}$ PF: $\{(f_1,f_2,f_3) \lvert f_1+f_2+f_3 = 1, f_i \in [0,1], i = 1,2,3\}$	Many boundary attractors Linear PF
MOP7	$f_1(x) = (1+g(x))\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = (1+g(x))\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = (1+g(x))\sin(0.5\pi x_1)$ $g(x) = 2\sin(\pi x_1)\sum_{i=3}^{n}(-0.9t_i^2 + \lvert t_i\rvert^{0.6})$ $t_i = x_i - x_1x_2,\ i = 3,\ldots,n$	PS: $\{x \in R^{(n-2)} \lvert x_i = x_1x_2, 2 \le i \le n, x_j \in [0,1], j = 1,2\}$ PF: $\{(f_1,f_2,f_3) \lvert f_1^2+f_2^2+f_3^2 = 1, f_i \in [0,1], i = 1,2,3\}$	Many boundary attractors Concave PF
MOP8	$f_1(x) = (1+g(x))x_1x_2$ $f_2(x) = (1+g(x))x_1(1-x_2)$ $f_3(x) = (1+g(x))(1-x_1)$ $g(x)=5(\lvert\cos(\pi x_1)\rvert+\lvert\cos(\pi x_2)\rvert)\sum_{i=3}^{n}\frac{\lvert t_i\rvert}{1+\exp(5\lvert t_i\rvert)}$ $t_i = x_i - x_1x_2,\ i = 3,\ldots,n$	PS: $\{x \in R^{(n-2)} \lvert x_i = x_1x_2, 2 \le i \le n, x_j \in [0,1], j = 1,2\}$ PF: $\{(f_1,f_2,f_3) \lvert f_1+f_2+f_3 = 1, f_i \in [0,1], i = 1,2,3\}$	One intermediate attractor Linear PF
MOP9	$f_1(x) = (1+g(x))\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = (1+g(x))\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = (1+g(x))\sin(0.5\pi x_1)$ $g(x) = 2\sin(0.5\pi(x_1+x_2))\sum_{i=3}^{n}(\lvert t_i\rvert^{0.6} - 0.9t_i^2)$ $t_i = x_i - x_1x_2,\ i = 3,\ldots,n$	PS: $\{x \in R^{(n-2)} \lvert x_i = x_1x_2, 2 \le i \le n, x_j \in [0,1], j = 1,2\}$ PF: $\{(f_1,f_2,f_3) \lvert f_1^2+f_2^2+f_3^2 = 1, f_i \in [0,1], i = 1,2,3\}$	Two extremal attractors Concave PF



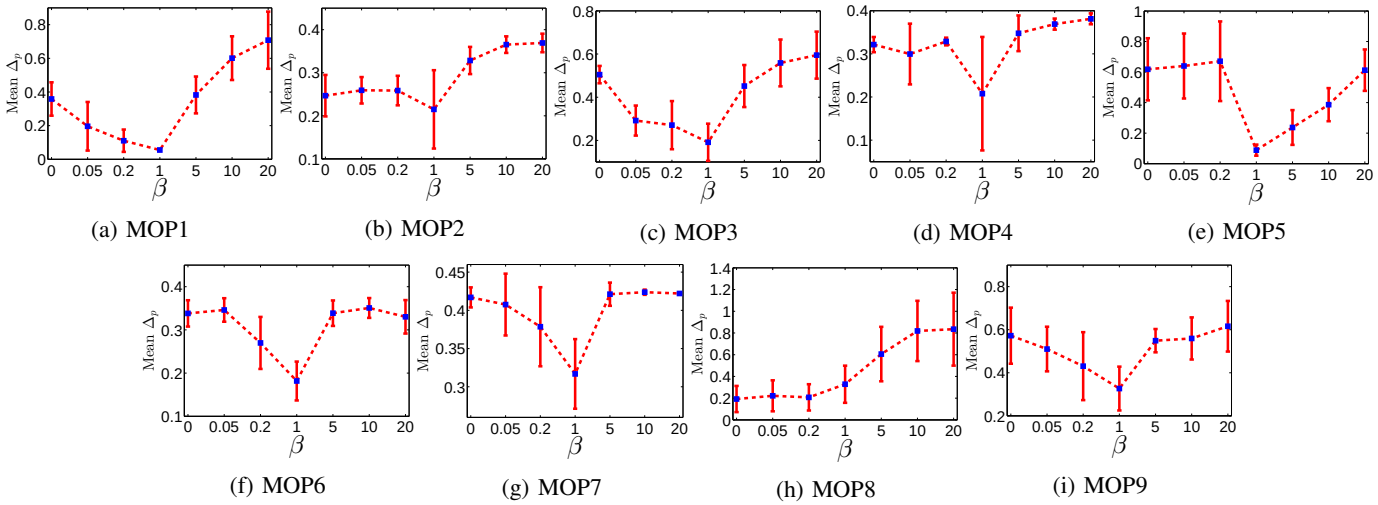
(a) MOP1 (b) MOP2 (c) MOP3 (d) MOP4 (e) MOP5

(f) MOP6 (g) MOP7 (h) MOP8 (i) MOP9

Fig. 22: Mean IGD values obtained by MSF with different β settings.

XIV. Results on a Small Population Size

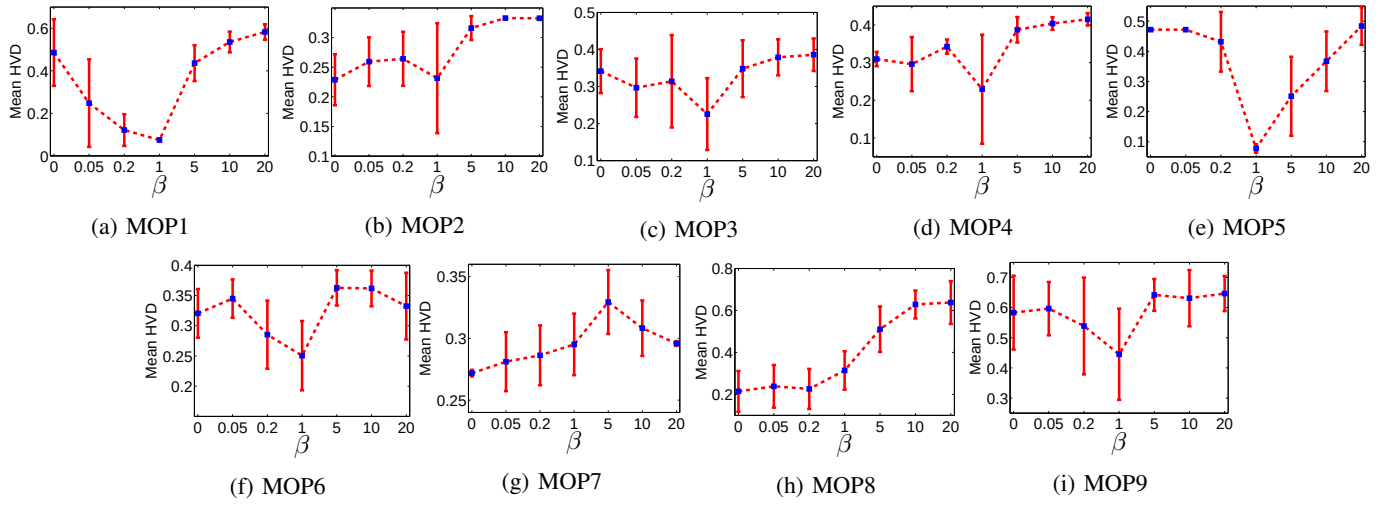
In previous experiments, a population size of no less than 100 is used, which means there are a high number of subproblems generated by scalarizing functions and neighboring subproblems are much similar. In this case, it may be not very



(a) MOP1 (b) MOP2 (c) MOP3 (d) MOP4 (e) MOP5

(f) MOP6 (g) MOP7 (h) MOP8 (i) MOP9

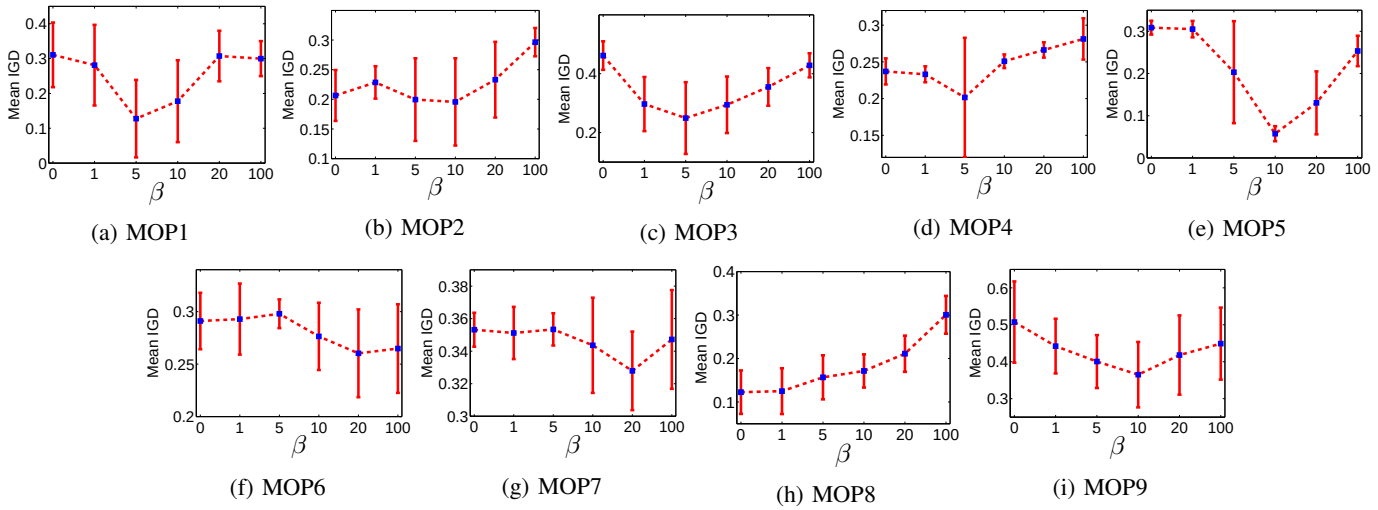
Fig. 23: Mean Δ_p values obtained by MSF with different β settings.



(a) MOP1 (b) MOP2 (c) MOP3 (d) MOP4 (e) MOP5

(f) MOP6 (g) MOP7 (h) MOP8 (i) MOP9

Fig. 24: Mean HVD values obtained by MSF with different β settings.

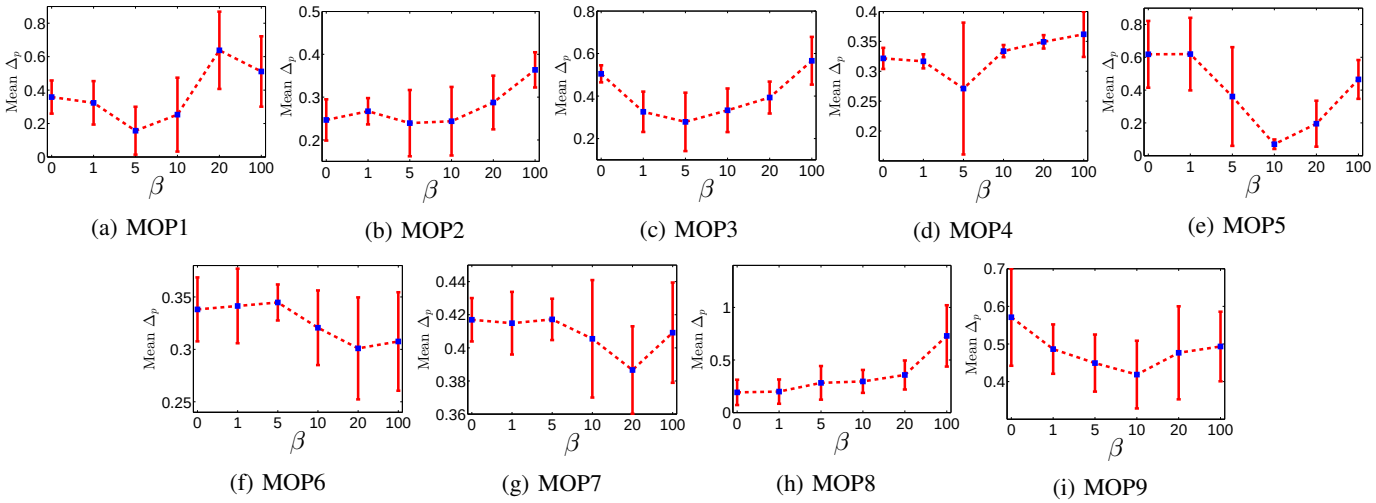


(a) MOP1 (b) MOP2 (c) MOP3 (d) MOP4 (e) MOP5

(f) MOP6 (g) MOP7 (h) MOP8 (i) MOP9

Fig. 25: Mean IGD values obtained by PSF with different β settings.

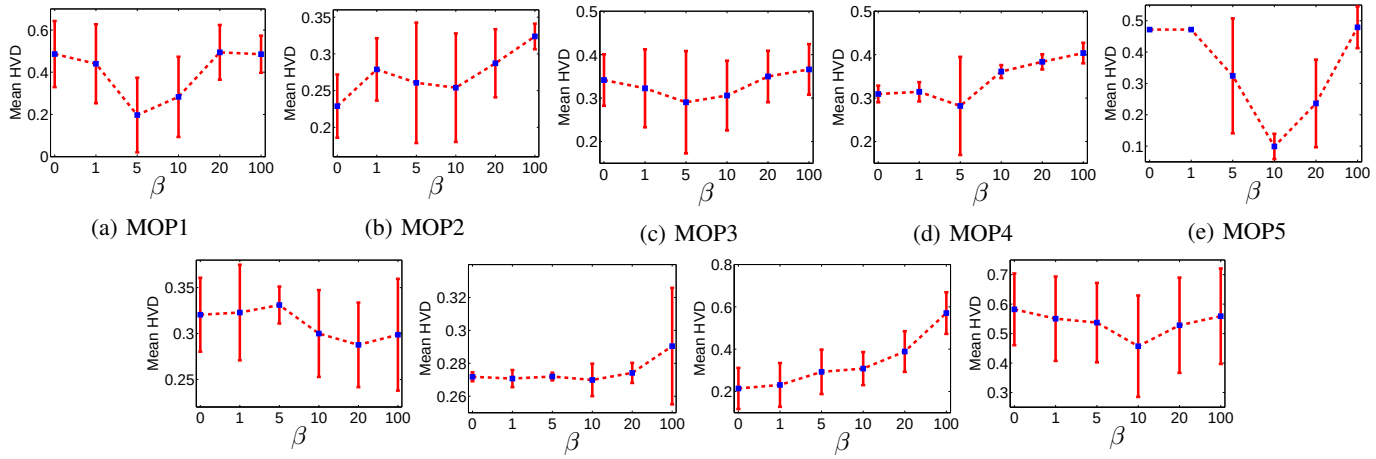
clear to see the difference between the proposed scalarizing functions and the other popular ones. For this reason, we would like to reduce the population size to 10, and the number of generations is equivalently changed to 50000 so



(a) MOP1 (b) MOP2 (c) MOP3 (d) MOP4 (e) MOP5

(f) MOP6 (g) MOP7 (h) MOP8 (i) MOP9

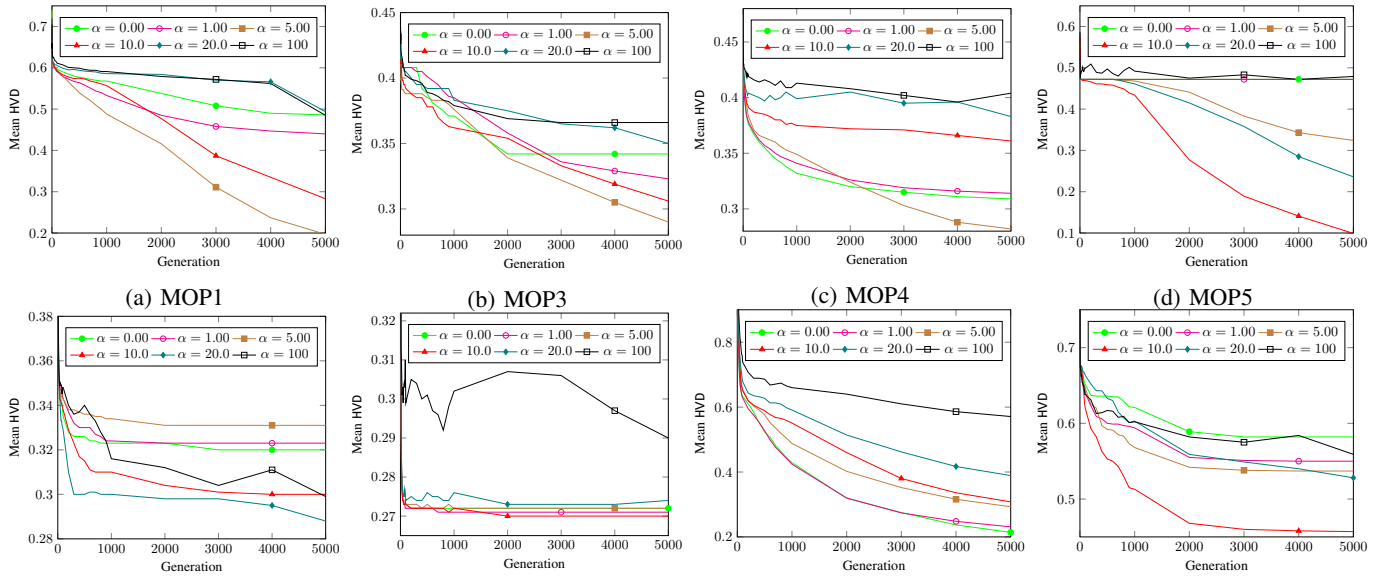
Fig. 26: Mean Δ_p values obtained by PSF with different β settings.



(a) MOP1 (b) MOP2 (c) MOP3 (d) MOP4 (e) MOP5

(f) MOP6 (g) MOP7 (h) MOP8 (i) MOP9

Fig. 27: Mean HVD values obtained by PSF with different β settings.



(a) MOP1 (b) MOP3 (c) MOP4 (d) MOP5

(e) MOP6 (f) MOP7 (g) MOP8 (h) MOP9

Fig. 28: Evolution curve of the mean HVD metric obtained by PSF with different β settings.

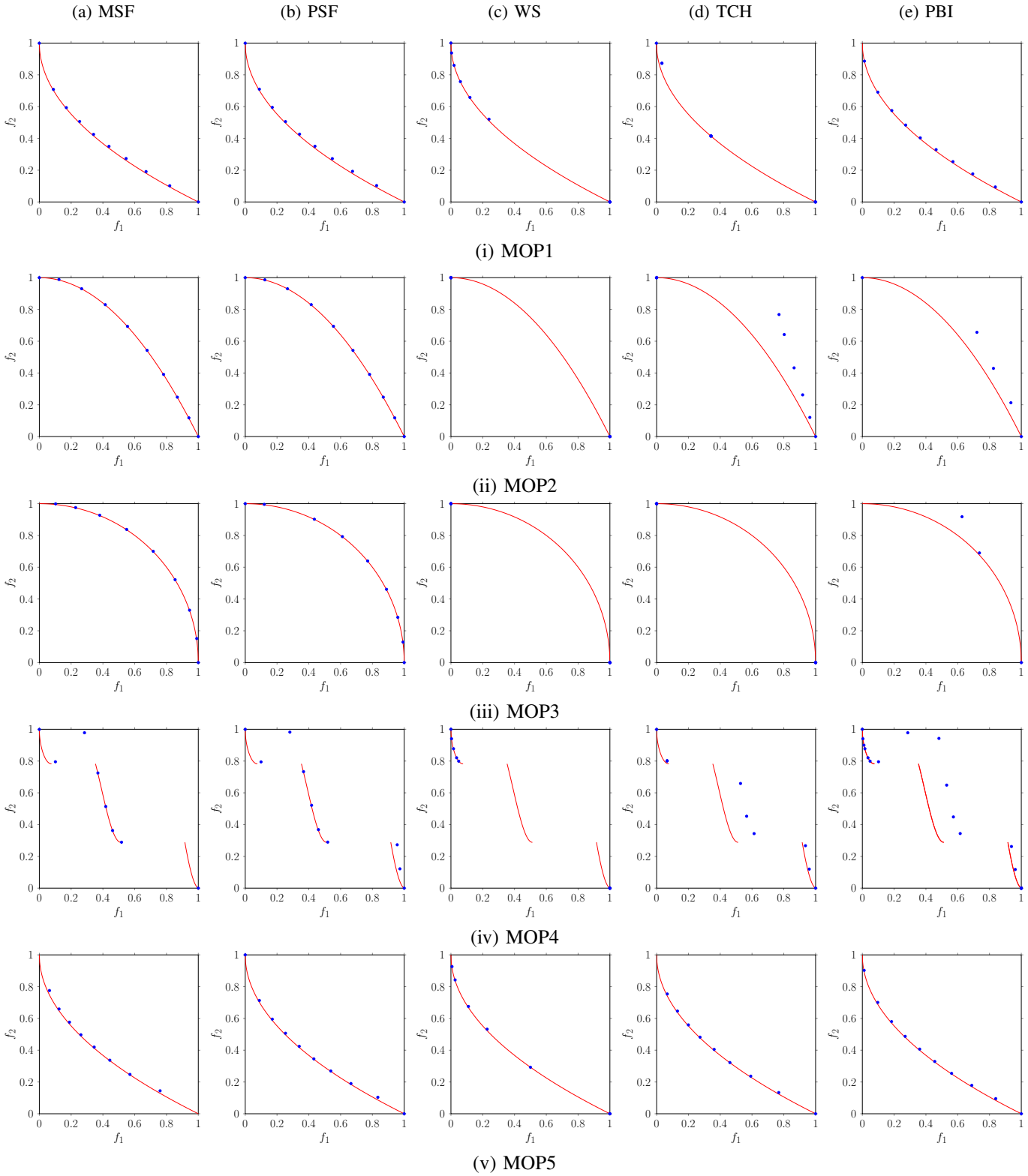
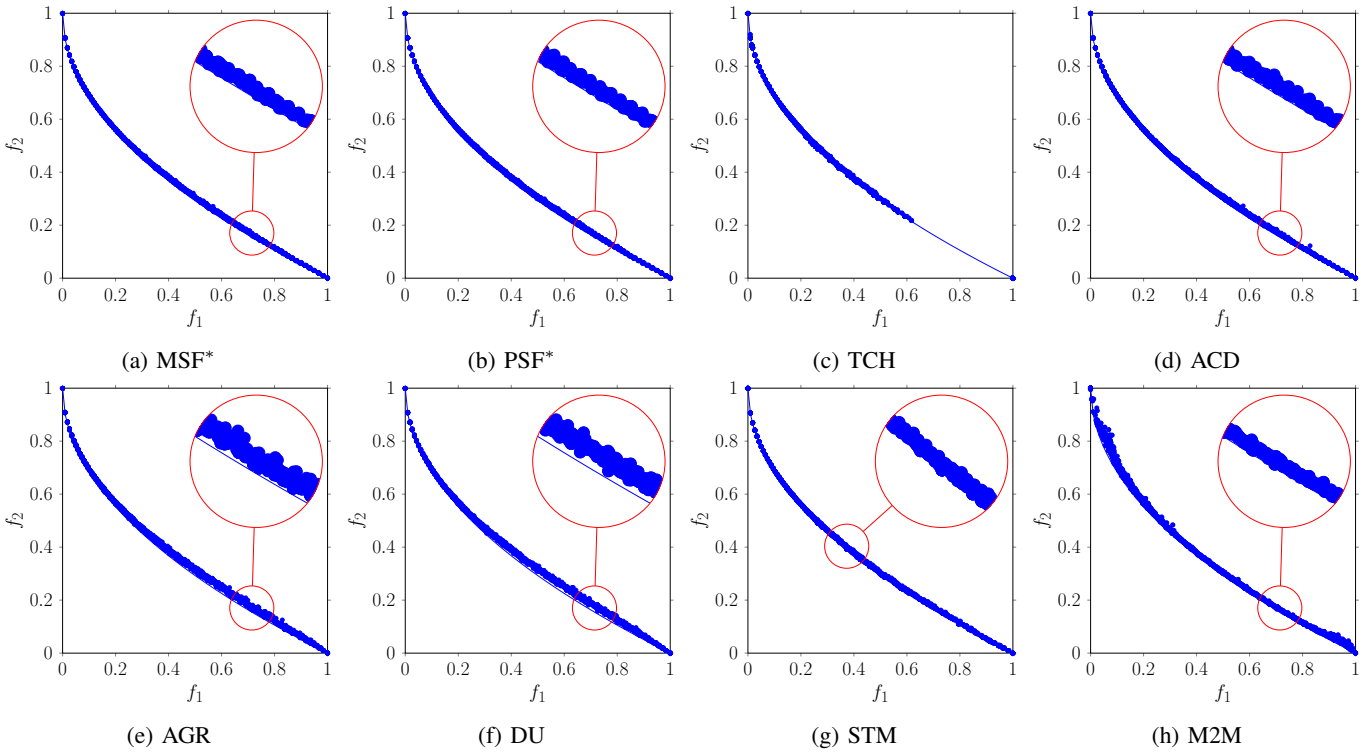


Fig. 29: Best PF approximations obtained by five scalarizing methods with a small population size.

that the number of function evaluations remains the same as our previous experiments. MSF with $\alpha = 1$ and PSF with $\alpha = 10$ are tested against the WS method, the TCH method and the PBI method on five MOP problems with different PF characteristics.

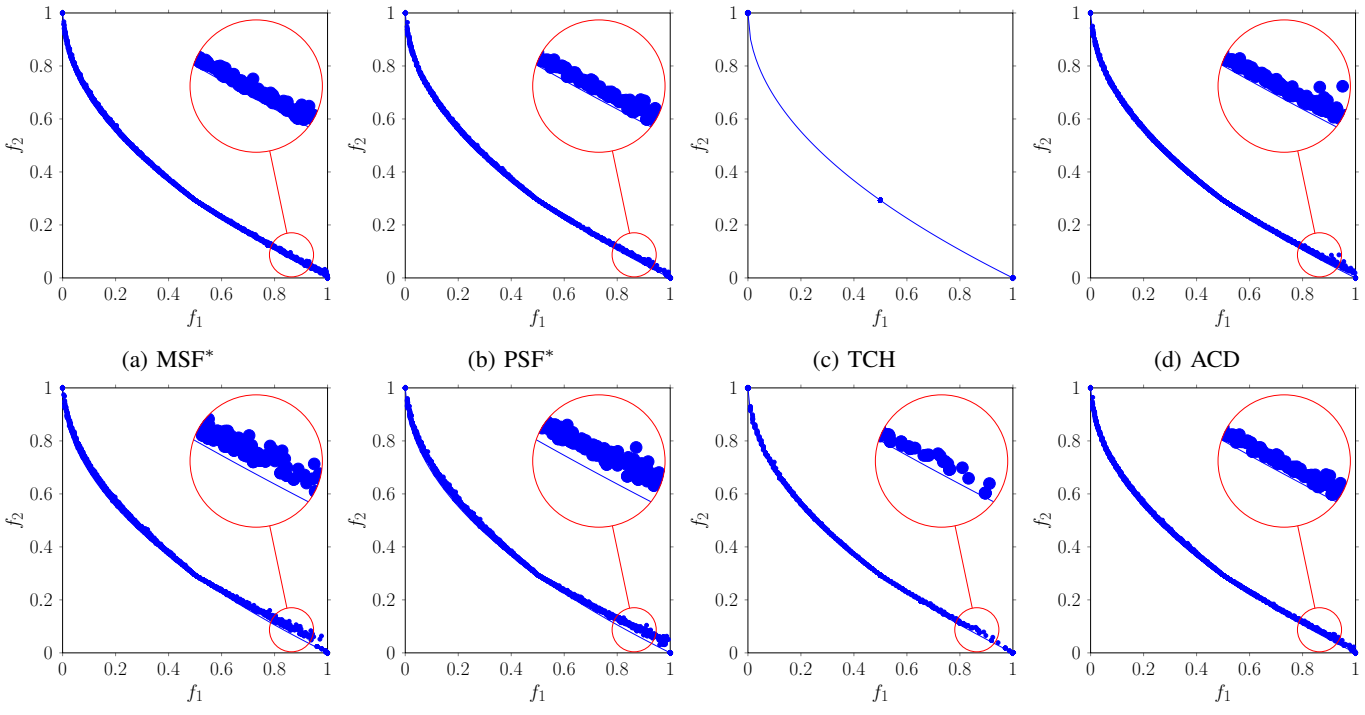
Fig. 29 presents the best PF approximations by five different scalarzing functions when a population size of 10 was used. It can be observed that both MSF and PSF can nicely approximate the PFs with different geometries. The WS method can obtain some good solutions when the PF is convex but performs poorly if the PF is concave. The TCH method has a nice performance on MOP5 but does not work on the other



(a) MSF* (b) PSF* (c) TCH (d) ACD

(e) AGR (f) DU (g) STM (h) M2M

Fig. 30: The whole PF approximations of MOP1 obtained by ten algorithms over 30 runs.



(a) MSF* (b) PSF* (c) TCH (d) ACD

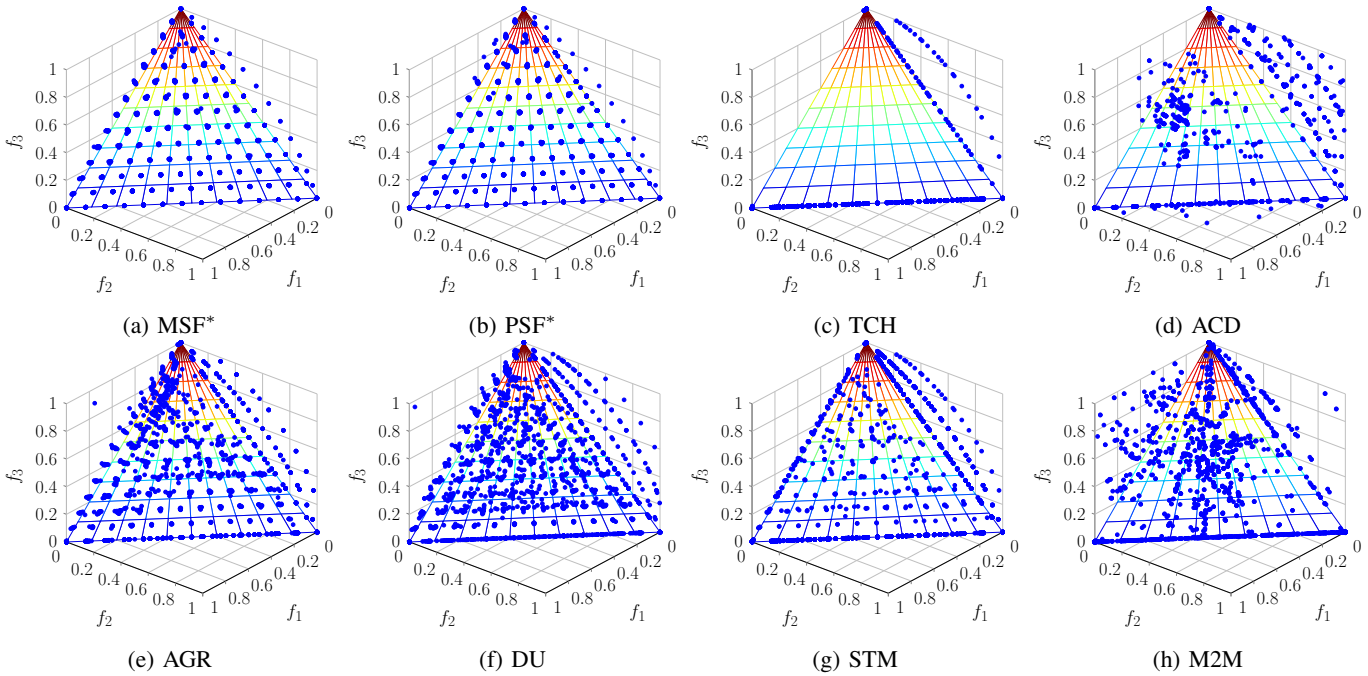
(e) AGR (f) DU (g) STM (h) M2M

Fig. 31: The whole PF approximations of MOP5 obtained by ten algorithms over 30 runs.

problems. The PBI method shows some promising results, but it seems to converge slowly as there a big gap between approximations and the PF for MOP2-4. Overall, the results are able to demonstrate that the new scalarizing functions are very promising and can be better options for solving the problems considered here.

XV. Visual PF Approximations

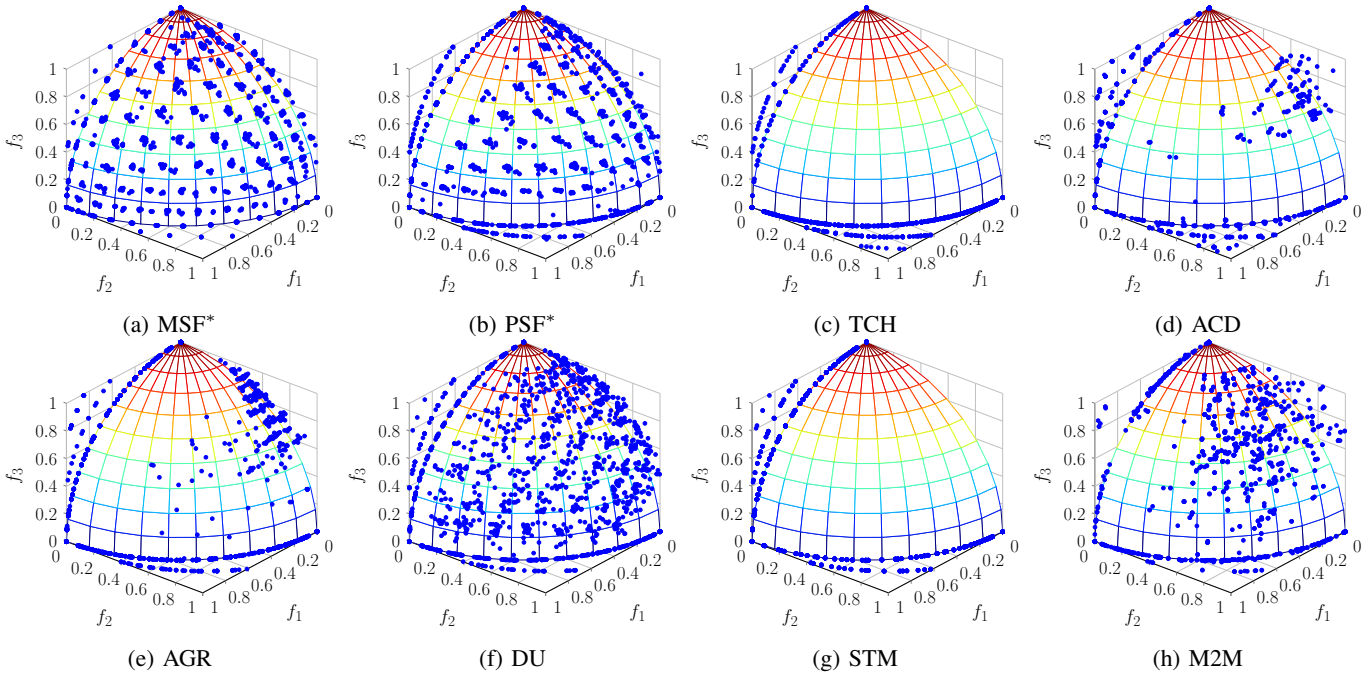
Figs. 30 to 33 show the PF approximations obtained by ten algorithms for four selected test problems. As can be seen from the figures, ACD, AGR, DU and M2M converge relatively slowly (observed from the two bi-objective problems) whereas TCH and STM cannot maintain diversity well (observed from



(a) MSF* (b) PSF* (c) TCH (d) ACD

(e) AGR (f) DU (g) STM (h) M2M

Fig. 32: The whole PF approximations of MOP6 obtained by ten algorithms over 30 runs.



(a) MSF* (b) PSF* (c) TCH (d) ACD

(e) AGR (f) DU (g) STM (h) M2M

Fig. 33: The whole PF approximations of MOP9 obtained by ten algorithms over 30 runs.

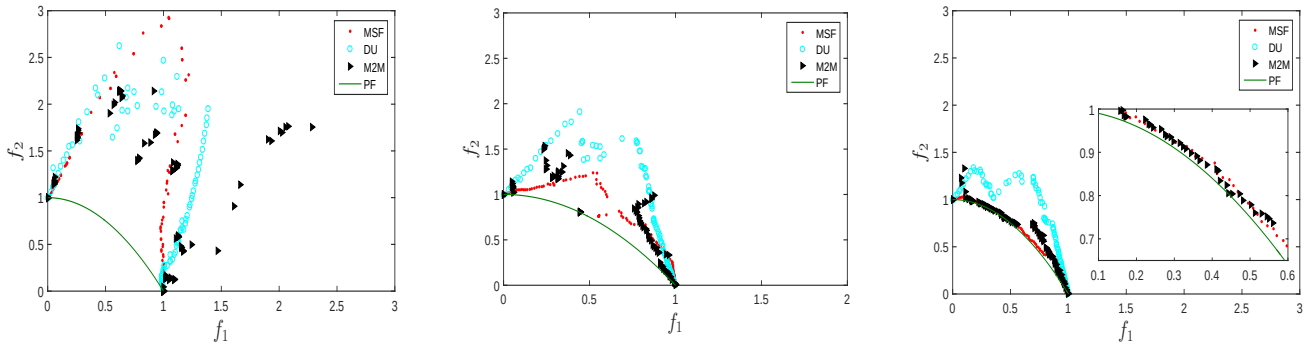
the triobjective MOP9). Nevertheless, both MSF* and PSF* show better performance compared with the other algorithms.

XVI. PF Approximations at Different Search Stages

Figs 34 and 35 show the PF approximations obtained by several selected algorithms for MOP2 and MOP3 after 50, 500, and 1000 generations. As can be seen, the MOP problems are hard to solve and their PF can be hardly approximated within 1000 generations. To obtain a good PF approximation, a high number of generations is needed, although it is less likely to be used in practice. However, if time is not the focus and sufficient computational resources are available, it is not trivial to use more generations in order to achieve high approximation accuracy. The figures also show that MSF can provide competitive and sometimes even better results compared with the other algorithms at three different stages. Thus, MSF performs well regardless of the number of generations, although a high number of generations is needed to guarantee the closeness of the algorithm to the PF in a good manner.

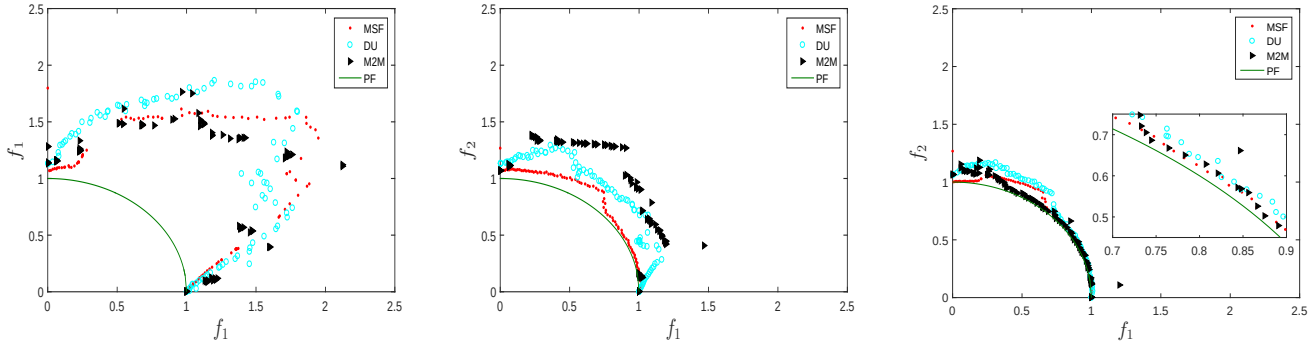
XVII. Influence of Recombination Operators

This section investigates the influence of different recombination operators on the performance of our methods. We have



(a) after 50 generations (b) after 500 generations (c) after 1000 generations

Fig. 34: PF approximations of three algorithms with the lowest IGD value for MOP2 after 50, 500 and 1000 generations.



(a) after 50 generations (b) after 500 generations (c) after 1000 generations

Fig. 35: PF approximations of three algorithms with the lowest IGD value for MOP3 after 50, 500 and 1000 generations.

compared Li and Liu's operator [3] (named LLX) with simulated binary crossover (SBX) [1] and differential evolution (DE) [2] on the problems MOP1 and MOP2. The investigation is conducted on the MSF* framework, and parameter settings remain the same (parameters in DE are same as those of MOEA/D-DE).

Table VIII provides the HV results obtained by different recombination operators. It is clear to see that the performance depends heavily on the recombination operator chosen for population reproduction. LLX performs best in terms of the HV metric, followed by DE, and SBX ranks the worst. For visual inspection, we also plot the whole PF approximations over 30 independent runs obtained by each recombination operator, which are shown in Figs. 36 and 37. It is clear again that LLX shows much better performance than the other two operators. A possible explanation for the high performance of LLX is that LLX can focus on generating diversified individuals at the early stage of the search and then place more emphasis on population convergence as the evolution proceeds. This kind of adaptive strategy helps yield appealing approximations.

TABLE VIII: Best, median and worst HV values obtained by MSF with different recombination operators

Problem	SBX	DE	LLX
MOP1	4.98E-02	3.26E-02	**1.39E-02**
	5.93E-02	3.75E-02	**1.43E-02**
	8.46E-02	4.25E-02	**1.50E-02**
MOP2	1.18E-01	1.02E-02	**6.30E-03**
	2.07E-01	7.35E-02	**6.39E-03**
	3.27E-01	3.27E-01	**6.53E-03**

REFERENCES

[1] K. Deb, S. Agrawwal, A. Pratap, and T. Meyarivan, " A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[2] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[3] H. Liu and X. Li, "The multiobjective evolutionary algorithm based on determined weight and sub-regional search," in *Proc. Congr. Evol. Comput.*, vol. 15, 2009, pp. 361–384.

[4] K. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA: Kluwer, 1999.

[5] A. J. Wilson, "Volume of n-dimensional ellipsoid," *Sciencia Acta Xaveriana*, vol. 1, no. 1, pp. 101-106, 2009.

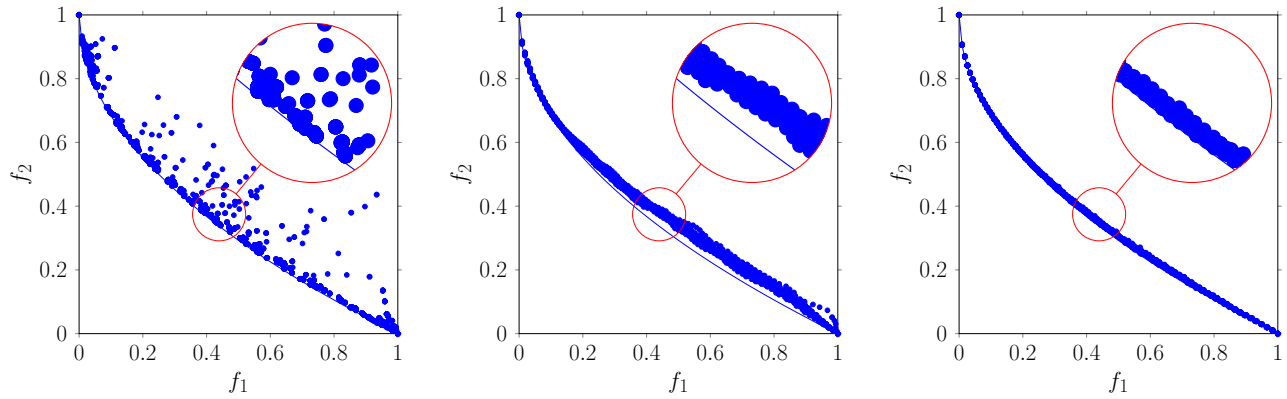


Fig. 36: The whole PF approximations of MOP1 obtained by different recombination operators over 30 runs.

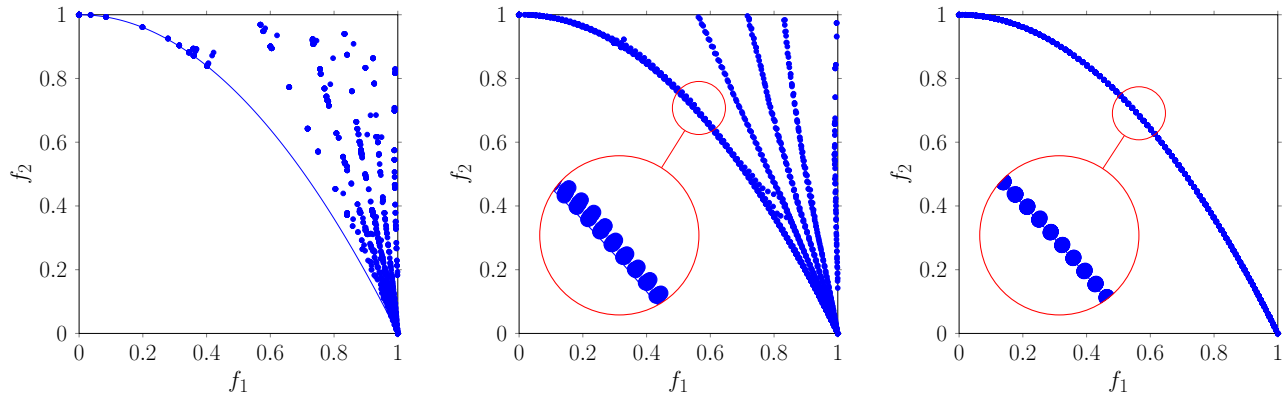


Fig. 37: The whole PF approximations of MOP2 obtained by different recombination operators over 30 runs.